

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

кафедра обчислювальної техніки

До захисту допущено:

Завідувач кафедри

Сергій СТИРЕНКО

«16» травня 2020 р.

Дипломний проект

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Комп'ютерні системи та мережі»
спеціальності 6.050102 «Комп'ютерна інженерія»
на тему: «Контролер керування автоматичним запуском побутових
електростанцій»**

Виконав:

студент IV курсу, групи ІО-361

Данил Євгенійович БАРАННИК

Керівник:

Асистент

Владислав Ігорович ТАРАН

Консультант з нормоконтролю:

Професор, доктор технічних наук

Валерій Павлович СІМОНЕНКО

Рецензент:

Доц. каф. АУТС к.т.н.

Андрій Володимирович ПИСАРЕНКО

Засвідчую, що у цьому дипломному
проекті немає запозичень з праць інших
авторів без відповідних посилань.

Студент Баранник Д.Є.

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 6.050102 «Комп'ютерна інженерія»

Освітньо-професійна програма «Комп'ютерні системи та мережі»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Сергій СТИРЕНКО

«16» травня 2020 р.

ЗАВДАННЯ

на дипломний проект студенту

Баранник Данилу Євгенійовичу

1. Тема проекту «Контролер керування автоматичним запуском побутових електростанцій», керівник проекту асистент Таран В.І., затверджені наказом по університету від «07» травня 2020 р. № 1081-с
2. Термін подання студентом проекту 16.05.2020р.
3. Вихідні дані до проекту технічна документація.
4. Зміст пояснювальної записки
Розділ 1 – Огляд агрегатів та автоматики.
Розділ 2 – Вимоги до розробляемого контролеру.
Розділ 3 – Проектування та розробка контролеру.
Розділ 4 – Системи для дистанційного керування генератором через GSM/GPRS.
Розділ 5 – Тестування всіх модулів на об'єкті.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо)

Діаграма методів.

Блок-схема алгоритму взаємодії.

Схема підключення контролера до бензогенератора.

Схема плати контролера у програмі PCAD 2006.

Код програми РКІ екрану.

Код програми контролера.

Код програми дистанційного керування генератором.

6. Дата видачі завдання 13.04.2020р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту	Строк виконання етапів проекту	Примітка
1	Затвердження теми роботи	13.04.2020	виконано
2	Вивчення та аналіз завдання	16.04.2020	виконано
3	Проведення огляду генераторів за можливістю створення контролера керування автоматичним запуском	17.04.2020-20.04.2020	виконано
4	Проведення аналізу засобів для розробки програмного забезпечення та плати контролера	20.04.2020-24.04.2020	виконано
5	Розробити програмну реалізацію контролера	24.04.2020-26.04.2020	виконано
6	Розробити плату контролера	26.04.2020-01.05.2020	виконано
7	Провести тестування на генераторі та аналіз даних	01.05.2020-12.05.2020	виконано
8	Оформлення матеріалів роботи	12.05.2020-16.05.2020	
8	Передзахист	29.05.2020	виконано
9	Захист		

Студент

Данил Євгенійович БАРАННИК

Керівник

Владислав Ігорович ТАРАН

Анотація

Робота присвячена розробці сучасного цифрового контролера, спроектованого з урахуванням сучасних вимог і технологій. Основне призначення - надавати чи заміщати блок автоматики для синхронних побутових бензинових генераторів з електростартером номінальною потужністю до 10 кВт. У цьому сегменті більшість агрегатів представлені без блоку автоматики, або він продається окремо, або має обмежений функціонал, або відсутній зовсім. Даний контролер є універсальним, тому підходить до будь-якої моделі бензогенераторів.

При зникненні напруги запропонований контролер забезпечує інтелектуальний автоматичний запуск генератора в залежності від його температури, перемикає споживачів з мережі на агрегат, а також коректне відключення агрегату при появі напруги в мережі. Тобто даний контролер також забезпечує функцію автоматичного введення резерву (АВР). До додаткових функцій можна віднести: управління заслінкою під час запуску генератора при температурі двигуна нижче заданої (T_g), контроль мотогодин, гнучке налаштування всіх основних параметрів, поліпшену функцію енергозбереження. Конструкція розробленого контролера дозволяє легко адаптувати його до будь-якого бензогенератора з електростартером. Передбачено можливість автоматичного і напівавтоматичного режимів дистанційного керування генератором зі збереженням можливості ручного керування. Використання даного контролеру виключає внесення змін до функціональності агрегату і у вимкненому стані контролер ніяк не впливає на його роботу.

Аннотация

Работа посвящена разработке современного цифрового контроллера, спроектированного с учетом современных требований и технологий. Основное назначение - предоставлять или замещать блок автоматики для синхронных бытовых бензиновых генераторов с электростартером номинальной мощностью до 10 кВт. В этом сегменте большинство агрегатов представлены без блока автоматики, или он продается отдельно, или имеет ограниченный функционал, или отсутствует вовсе. Данный контроллер является универсальным, поэтому подходит к любой модели бензогенераторов.

При исчезновении напряжения предложенный контроллер обеспечивает интеллектуальный автоматический запуск генератора в зависимости от его температуры, переключение потребителей с сети на агрегат, а также корректное отключение агрегата при появлении напряжения в сети. То есть данный контроллер также обеспечивает функцию автоматического ввода резерва (АВР). К дополнительным функциям можно отнести: управление заслонкой при запуске генератора при температуре двигателя ниже заданной (T_g), контроль моточасов, гибкая настройка всех основных параметров, улучшенную функцию энергосбережения. Конструкция разработанного контроллера позволяет легко адаптировать его к любому бензогенератору с электростартером. Предусмотрена возможность автоматического и полуавтоматического режимов дистанционного управления генератором с сохранением возможности ручного управления. Использование данного контроллера исключает внесения изменений в функциональности агрегата и в выключенном состоянии контроллер никак не влияет на его работу.

Abstract

The work is devoted to the development of a modern digital controller, designed taking into account modern requirements and technologies. The main purpose is to provide or replace an automation unit for synchronous household gasoline generators with an electric starter with a rated power of up to 10 kW. In this segment, most generators are presented without an automation unit, or it is sold separately, or has limited functionality, or is completely absent. This controller is universal, therefore suitable for any model of gas generators.

In case of voltage disappearance, the proposed controller provides intelligent automatic start of the generator depending on its temperature, switching consumers from the network to the unit, as well as the correct shutdown of the unit when voltage appears in the network. This controller also provides an automatic voltage regulator (AVR) function. Additional functions include: control of the damper when starting the generator at an engine temperature below a predetermined temperature (T_g), control of engine hours, flexible adjustment of all basic parameters, and an improved energy-saving function. The design of the developed controller allows you to easily adapt it to any gas generator with an electric starter. The possibility of automatic and semi-automatic modes of remote control of the generator with the preservation of the possibility of manual control. Using this controller excludes changes in the functionality of the unit and in the off state the controller does not affect its operation.

Технічне завдання до дипломного проекту

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ	2
2. ПІДСТАВИ ДЛЯ РОЗРОБКИ	2
3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ	2
4. ДЖЕРЕЛА РОЗРОБКИ	2
5. ТЕХНІЧНІ ВИМОГИ	3
5.1. Вимоги до розробленого продукту	3
5.2. Вимоги до програмного забезпечення	4
5.3. Вимоги до апаратного забезпечення	4

					ІАЛЦ.467100.002 ТЗ			
Зм.	Арк.	№ докум.	Підпис	Дата	Контролер керування автоматичним запуском побутових електростанцій Технічне завдання	Літ.	Аркуш	Аркушів
Розробив		Баранник Д.Є.					1	4
Перевір.		Таран В. І.						
Н. контр.		Сімоненко В.П.				КПІ ім. Ігоря Сікорського Каф. ФІОТ, гр. ІО-361		
Затверд.								

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Дане технічне завдання розповсюджується на розробку контролера, призначеного надавати чи заміщати блок автоматики для синхронних побутових бензинових генераторів з електростартером, номінальною потужністю до 10 кВт.

Область застосування: розробка пристроїв керування автоматичним запуском побутових електростанцій.

2. ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки служить завдання на виконання розробки контролера керування автоматичним запуском електростанцій, затвердженою кафедрою обчислювальної техніки Національного технічного Університету України «Київський Політехнічний Інститут».

3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою даного проекту є розробка контролера керування автоматичним запуском електростанцій.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелами для розробки служать науково-технічна література з комп'ютерних технологій, з теорії і практики програмування, публікації в Інтернеті з даних питань.

					ІАЛЦ.467100.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		2

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до розробленого продукту

- Інтелектуальне управління запуском агрегату в залежності від його температури і інших параметрів;
- Індикацію параметрів, режимів роботи і аварійних ситуацій з їх повної розшифровкою на РКІ;
- Продуманий інтерфейс російською мовою;
- Гнучке налаштування всіх основних параметрів;
- Функції енергозбереження (0.3 Вт - в режимі простою);
- Контроль наробки агрегату;
- Можливість оновлення програмного забезпечення;
- Дистанційне керування виконавчим блоком.

Контролер повинен забезпечувати виконання наступних функцій:

- Запуск генератора в автоматичному режимі роботи при зниженні напруги нижче допустимого рівня $U_c < 175V$ (80%);
- Управління заслінкою (ЕПЗ) під час запуску генератора при температурі двигуна нижче заданої (T_g);
- Прогрівання двигуна, якщо його температура нижче заданої (T_n);
- Автоматичний перехід живлення навантаження на мережу і останов генератора в автоматичному режимі роботи при відновленні напруги мережі;
- Охолодження двигуна на холостому ходу;
- Прийом навантаження в разі зникнення напруги мережі при охолодженні;
- Напіваавтоматичний запуск генератора з відключенням мережі і включенням живлення навантаження від генератора;
- Напіваавтоматичний останов генератора з відключенням живлення навантаження від генератора і включенням мережі;
- Захист лінії генератора від перевантаження і короткого замикання;

					ІАЛЦ.467100.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

- Захист навантаження від високої напруги мережі $U_c > 264V$ (120%);
- Екстрений останів генератора без охолодження якщо натиснута кнопка аварійної зупинки;
- Індикацію основних параметрів: заряду акумулятора, напруги мережі, температури і напруцювання агрегату;
- Видачу аварійних попереджень:
 - при відсутності запуску агрегату;
 - при розряді акумуляторної батареї;
 - при перевищенні допустимого напруги мережі;
 - при відсутності струму, що виробляється генератором (при його роботі).

5.2. Вимоги до програмного забезпечення

Для роботи контролера програмне забезпечення не потребується.

5.3. Вимоги до апаратного забезпечення

Для можливості оновлення програмного забезпечення необхідно наступне:

- Atmel Studio 7;
- Windows XP/7/Vista/8/10, або MacOS, або Linux;
- USB 2.0+.

					ІАЛЦ.467100.002 ТЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	3
ВСТУП	4
Актуальність теми	4
Мета і задачі розробки	4
РОЗДІЛ 1	5
ОГЛЯД АГРЕГАТИВ ТА АВТОМАТИКИ	5
1.1. Історія електроагрегатів	6
1.1.1. Історія електричного генератора	7
1.2. Різновиди бензинових електростанцій	9
1.2.1. За мобільністю	9
1.2.2. За конструктивним виконанням	10
1.2.3. За родом струму	10
1.2.4. За типом генератора змінного струму	11
1.3. Інверторні генератори	12
1.4. Час роботи генератора	13
1.5. Спосіб запуску генератора	14
1.6. Призначення бензинових електростанцій та автоматики	14
1.7. Автоматика до генераторів	16
ВИСНОВКИ ДО РОЗДІЛУ 1	19
РОЗДІЛ 2	20
ВИМОГИ ДО РОЗРОБЛЯЄМОГО КОНТРОЛЕРУ	20
2.1. Вимоги до програмного забезпечення	22
2.2. Вимоги до технічної характеристики плати	22
ВИСНОВКИ ДО РОЗДІЛУ 2	24
РОЗДІЛ 3	25
ПРОЕКТУВАННЯ ТА РОЗРОБКА КОНТРОЛЕРУ	25

					ІАЛЦ.467100.003 ПЗ			
Зм.	Арк.	№ докум.	Підпис	Дата				
Розробив		Баранник Д.Є.			Контролер керування автоматичним запуском побутових електростанцій Пояснювальна записка	Літ.	Арк.	Аркушів
Перевір.		Таран В. І.					1	67
						КПІ ім. Ігоря Сікорського		
Н. контр.		Сімоненко В.П.				Каф. ФІОТ, гр. ІО-361		
Затверд.								

3.6. Налаштування параметрів	36
3.7. Алгоритм роботи виконавчого блоку	38
3.8. Аварійна зупинка генератора	43
3.9. Рекомендовані параметри	43
3.10. Повідомлення про помилки	44
3.11. Створення прошивки	46
ВИСНОВКИ ДО РОЗДІЛУ 3	50
РОЗДІЛ 4	51
СИСТЕМИ ДЛЯ ДИСТАНЦІЙНОГО КЕРУВАННЯ ГЕНЕРАТОРОМ ЧЕРЕЗ GSM/GPRS	51
4.1. Опис роботи системи	51
4.2. Перелік необхідного	52
4.3. Алгоритм роботи системи	53
4.4. Підготовка та збірка	56
4.5. Зберігання та транспортування	57
ВИСНОВКИ ДО РОЗДІЛУ 4	58
РОЗДІЛ 5	59
ТЕСТУВАННЯ ВСІХ МОДУЛІВ НА ОБ'ЄКТІ	59
ВИСНОВКИ ДО РОЗДІЛУ 5	65
ВИСНОВКИ	66
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	67

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ATS	автоматика до генератора
ABP	автоматичне введення резерву
ККД	коефіцієнт корисної дії
ЕРС	електрорушійна сила
ЕПЗ	електропривід повітряної заслінки
GSM	глобальний стандарт цифрового мобільного стільникового зв'язку.
GPRS	General Packet Radio Service
TDMA	Time Division Multiple Access
FDMA	Frequency Division Multiple Access
MOVE	Models, Operations, Views, Events
API	Application Programming Interface
ПДС	Пристрій диференціального струму

ВСТУП

Контролер призначений надавати чи заміщати блок автоматики (ATS та АВР) для синхронних побутових бензинових генераторів з електростартером номінальною потужністю до 10 кВт, забезпечує інтелектуальний автоматичний запуск бензоелектричного агрегату при зникненні напруги в мережі, автоматичне введення резерву (АВР), перемикання споживачів з мережі на агрегат, а також коректне відключення агрегату при появі напруги в мережі. Пристрій може бути адаптоване для підключення до будь-якого бензогенераторів, що запускається електростартером.

Актуальність теми

Більшість побутових агрегатів потужністю до 10 кВт представлені без блоку автоматики, або він продається окремо, або має обмежений функціонал, або відсутній зовсім.

Мета і задачі розробки

Метою роботи є розробка універсального контролера, що дозволить користувачам керувати та контролювати власні генератори, забезпечувати свої об'єкти безпечнішим джерелом електроенергії.

Для досягнення поставленої мети були поставлені наступні основні задачі:

- Провести аналіз існуючих бензинових генераторів та підтвердити зможу підключення пропонованого контролеру;
- Створити плату та програмну реалізацію розробляемого контролера;
- Провести підключення і тестування на бензоелектричному агрегаті;
- Дослідити отриманні результати.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

РОЗДІЛ 1

ОГЛЯД АГРЕГАТИВ ТА АВТОМАТИКИ

Бензоелектричний агрегат (бензогенератор, бензинова електростанція) - рухома або стаціонарна енергетична установка, обладнана одним або декількома електричними генераторами з приводом від бензинового двигуна внутрішнього згоряння. Також існують електростанції з приводом від дизельного та газопоршневі електростанції. Як правило, бензоелектричний агрегат об'єднує в собі генератор змінного струму [1] і двигун внутрішнього згоряння [2], які встановлені на сталевій рамі, а також систему контролю і управління установкою.

Терміни бензоелектричний агрегат, бензогенератор, бензинова електростанція не є синонімами:

- бензогенератор - пристрій, що складається з конструктивно об'єднаних бензинового двигуна і генератора;
- бензоелектричний агрегат в свою чергу включає в себе бензогенератор, а також допоміжні пристрої: раму, прилади контролю, паливний бак;
- бензинова електростанція - це стаціонарна або пересувна установка на базі бензоелектричного агрегату, додатково включає в себе пристрої для розподілу електроенергії, пристрої автоматики, пульт управління.

Двигун внутрішнього згоряння надає руху синхронному [3] або асинхронному електричному генератору [4]. З'єднання двигуна і електричного генератора виробляється або безпосередньо фланцем, або через демпферну муфту. У першому випадку використовується двухопорний генератор, тобто генератор, який має два опорних підшипника, а в другому - з одним опорним підшипником (одноопорний).

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

1.1. Історія електроагрегатів

Електрична енергія довгий час була лише об'єктом експериментів і не мала практичного застосування. Перші спроби корисного використання електрики були зроблені в другій половині XIX століття, основними напрямками використання були винайдений телеграф, гальванотехніка, військова техніка (наприклад були спроби створення судів і самохідних машин з електричними двигунами; розроблялися міни з електричним детонатором). Джерелами електрики спочатку служили гальванічні елементи. Суттєвим проривом в масовому розповсюдженні електроенергії стало винахід електромашинних джерел електричної енергії - генераторів. У порівнянні з гальванічними елементами, генератори володіли більшою потужністю і ресурсом корисного використання, були істотно дешевше і дозволяли довільно задавати параметри виробленого струму. Саме з появою генераторів стали з'являтися перші електричні станції та мережі (до того джерела енергії були безпосередньо в місцях її споживання) - електроенергетика ставала окремою галуззю промисловості.

Першою в історії лінією електропередачі (в сучасному розумінні) стала лінія Лауфен - Франкфурт, яка заробила в 1891 році. Протяжність лінії становила 170 км, напруга 28,3 кВ, передана потужність 220 кВт [5]. У той час електрична енергія використовувалася в основному для освітлення в великих містах. Електричні компанії склалися в серйозної конкуренції з газовими: електричне освітлення перевершувало газове по ряду технічних параметрів, але було в той час істотно дорожче. З удосконаленням електротехнічного обладнання і збільшенням ККД генераторів, вартість електричної енергії знижувалася, і врешті-решт електричне освітлення повністю витіснило газове. Попутно з'являлися нові сфери застосування електричної енергії: удосконалювалися електричні підйомники, насоси та електродвигуни.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

Важливим етапом став винахід електричного трамвая: трамвайні системи були великими споживачами електричної енергії та стимулювали нарощування потужностей електричних станцій. У багатьох містах перші електричні станції будувалися разом з трамвайними системами.

1.1.1. Історія електричного генератора

Електричний генератор - пристрій, в якому неелектричні види енергії (механічна, хімічна, теплова) перетворюються в електричну енергію.

У 1827 угорський фізик Аньош Іштван Єдлик почав досліджувати електромагнітні обертові пристрої, які він назвав електромагнітними саморегуляторами.

У 1831 році Майкл Фарадей відкрив принцип роботи електромагнітних генераторів. Принцип, який назвали законом Фарадея, полягав в тому, що різниця потенціалів утворювалася між кінцями провідника, який рухався перпендикулярно магнітному полю. Він також побудував перший електромагнітний генератор, названий «диском Фарадея», який був уніполярним генератором, що використав мідний диск, що обертається між полюсами подковообразного магніту. Він виробляв невелику постійну напругу і сильний струм.

Динамо-машина стала першим електричним генератором, здатним виробляти потужність для промисловості. Її робота полягає на законах електромагнетизму для перетворення механічної енергії в пульсуючий постійний струм. Постійний струм вироблявся завдяки використанню механічного комутатора. Першу динамо-машину побудував Іполит Піксі (Ипполит Пикси, Hippolyte Pixii) в 1832 році.

Перший генератор електричного струму, заснований на явищі електромагнітної індукції, був побудований в 1832 році паризькими техніками братами Піксі. Генератор був забезпечений пристроєм для випрямлення струму.

Російський вчений Ленц Емілій Христіановіч ще 1833 році вказав на оборотність електричних машин: одна і та ж машина може працювати як

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

електродвигун, якщо її живити струмом, і може служити генератором електричного струму, якщо її ротор привести в обертання будь-яким двигуном, наприклад паровою машиною . У 1838 році Ленц, один з членів комісії з випробування дії електричного мотора Якобі, на досвіді довів оборотність електричної машини.

У 1843 році, був побудований генератор Еміля Штерера. У цієї машини було три сильних рухливих магніти і шість котушок, що оберталися від рук навколо вертикальної осі.

Таким чином, на першому етапі розвитку електромагнітних генераторів струму (до 1851 року) для отримання магнітного поля застосовували постійні магніти. На другому етапі (1851-1867 рр.) створювалися генератори, у яких для збільшення потужності постійні магніти були замінені електромагнітами. Їх обмотка живилась струмом від самостійного невеликого генератора струму з постійними магнітами. Подібна машина була створена англійцем Генрі Уальдом в 1863 році. При експлуатації цієї машини з'ясувалося, що генератори, забезпечуючи електроенергією споживача, можуть одночасно жити струмом і власні магніти. Виявилося, що сердечники електромагнітів зберігають залишковий магнетизм після виключення струму. Завдяки цьому генератор з самозбудженням дає струм і тоді, коли його запускають зі стану спокою. У 1866-1867 роках ряд винахідників отримали патенти на машини з самозбудженням.

У 1870 році бельгієць Зеноб Грам, який працював у Франції, створив генератор, який отримав широке застосування в промисловості. У своїй динамо-машині він використовував принцип самозбудження і вдосконалив кільцевої якір, винайдений ще в 1860 році А. Пачінотті.

До того, як було відкрито зв'язок між електрикою і магнетизмом, використовувалися електростатичні генератори, які працювали на основі принципів електростатики. Вони могли виробляти високу напругу, але мали маленький струм. Їх робота була заснована на використанні наелектризованих ременів, пластин і дисків для перенесення електричних зарядів з одного

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

електрода на інший. Заряди вироблялися, використовуючи один з двох принципів:

- електростатичну індукцію;
- трибоелектричний ефект, при якому електричний заряд виникав через механічний контакт двох діелектриків.

1.2. Різновиди бензинових електростанцій

Бензинові електростанції розрізняються по вихідній електричній потужності, виду струму (змінний трифазний / однофазний, постійний), вихідній напрузі, а також частоті струму (наприклад, 50, 60, 400 Гц). Бензинові електростанції поділяють за типом охолодження бензинового двигуна, - повітряного або рідинного. Електростанції з бензиновим двигуном рідинного охолодження - це агрегати великих потужностей і розмірів.

1.2.1. За мобільністю

Пересувні - електростанції потужністю, як правило, до 1000 кВт. Застосовуються в якості переносного (портативні) або резервного джерела електропостачання. Найчастіше представлені в спеціальному шумозахисному кожусі або контейнері зі стандартними (дозволеними) транспортувань габаритами. Стаціонарні (промислові) - електростанції, будь-якої потужності і типу, інтегровані в єдину систему енергокомплексу.

1.2.2. За конструктивним виконанням

- Відкритого виконання - базове виконання електростанції, призначене для розміщення електроустановки в спеціально обладнаному приміщенні;
- У шумозахисних кожусі - для установки в приміщення або на вулиці при наявності вимог до зниження шуму;
- Контейнерні - монтаж електростанції в блок-контейнер здійснюється для експлуатації установки в важких кліматичних умовах і підвищеної вандалозахищеності;

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

- Електростанція може бути встановлена в фургон, машину або на шасі. Таким чином, вона набуває статусу мобільного електростанції.

1.2.3. За родом струму

- Малопотужні бензинові електростанції виробляють, як правило, однофазний змінний струм напругою 220 В і / або трифазний напругою 380 В;
- Трифазні електростанції мають більш високий ККД за рахунок більш високого ККД генератора змінного струму;
- Переносні бензинові електростанції з вбудованим випрямлячем (інвертором) можуть мати додатковий вихід постійного струму напругою 12-14 вольт, наприклад, для зарядки акумуляторів;
- Потужні бензинові електростанції виробляють трифазний струм:
 - низьковольтні - з напругою до 1 кВ;
 - високовольтні - з напругою понад 1 кВ (6,3 кВ, 10 кВ).

Якщо необхідно передавати електроенергію, вироблену низьковольтними електростанціями, на значні відстань по лініях електропередачі, напруга підвищується на електричних підстанціях до 6,3 кВ або 10,5 кВ.

1.2.4. За типом генератора змінного струму

1. Синхронний генератор змінного струму.

Так як частота змінного струму синхронного генератора визначається числом обертів ротора (двигуна), то бензинова електростанція повинна мати механізм, що забезпечує постійне число обертів бензинового двигуна незалежно від навантаження (генерованої електричної потужності). Частота змінного струму синхронного генератора буде:

$$f = \frac{n}{60}, \quad (1.1)$$

де f - частота в герцах; n - число обертів ротора в хвилину.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

Якщо генератор має число пар полюсів ρ , то відповідно до цього частота електрорушійної сили такого генератора буде в ρ раз більше частоти електрорушійної сили двополюсного генератора:

$$f = \rho \frac{n}{60}. \quad (1.2)$$

ЕРС синхронного генератора регулюється зміною струму збудження.

2. Асинхронний генератор змінного струму.

Асинхронний генератор може генерувати змінний струм довільної, нестандартної частоти (значно відрізняється, наприклад, від використовуваної в промисловості та побуті частоти 50 Гц). Змінний струм після виходу з генератора піддається випрямлянню, потім отриманий постійний струм інвертор перетворює в змінний струм з параметрами, визначеними стандартом. Слід зазначити, що недорогі моделі інверторів мають на виході змінний струм несинусоїдної форми, зазвичай прямокутні імпульси або модифікована синусоїда.

ЕРС асинхронного генератора регулюється зміною числа обертів двигуна і зміною струму збудження (якщо передбачено конструкцією генератора).

1.3. Інверторні генератори

Інверторний генератор - електростанція в якій використовується інверторная система з регулятором широтно-імпульсної модуляції (ШІМ) для більш високої якості електроенергії (стабільність вихідної напруги і частоти). Такі електростанції ще називають «цифровими електрогенераторами», тому що оснащені електронними схемами управління.

Інверторний електрогенератор використовується для підключення електроніки або чутливої техніки за відсутності центрального електропостачання.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

Інверторну електростанцію можна сміливо рекомендувати як резервний генератор для дачі, так і для подорожей, дрібних ремонтних робіт. Види інверторних бензинових генераторів:

- інверторні портативні бензинові генератори потужністю до 1 кВт - можуть виконуватися у вигляді невеликого валізки, зручного для домашнього використання або при транспортуванні, використовуються при малої інтенсивності споживання електроенергії в домашніх або заміських умовах;
- інверторні бензинові генератори потужністю до 6 кВт - можуть виконуватися у вигляді моноблока з коліщатами для зручності в транспортуванні або вмонтовані в раму, використовуються при середній інтенсивності енергоспоживання і як резервні джерела при нетривалих перебоях в електропостачанні;
- бензинові інверторні генератори потужністю 10 кВт - для інтенсивної експлуатації в професійних цілях.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

На рис. 1.1. зображені приклади інверторних бензинових генераторів.



Рис. 1.1. Установки генераторні бензинові інверторні: зліва - потужністю 900 Вт, праворуч - потужністю 1700 Вт.

1.4. Час роботи генератора

Тривалість безперебійної роботи генератора залежить як від виду системи охолодження, так і від розмірів бензобака і інтенсивності навантаження. Чим більше приладів підключено до генератора, тим більше витрата бензину і тим менше часу буде безупинно функціонувати агрегат. Переносні бензинові електростанції малої потужності можуть працювати без дозаправки і відключення від одного до декількох годин. Цього часу цілком достатньо в польових умовах або під час відключення електроенергії на дачі. Потужні генератори мають паливні баки від 20-ти літрів і можуть безперервно працювати до 20-ти годин.

1.5. Спосіб запуску генератора

Відомо два варіанти запуску генератора:

1. Ручний запуск. Пристрої з ручним запуском найбільш поширені в силу своєї дешевини, простоти пристрою і надійності. Однак при необхідності запустити генератор на вулиці при низьких температурах вони не дуже зручні. Такі агрегати неможливо запустити дистанційно зі збереженням гарантії і заводської безпеки генератора.
2. Електрозапуск. Генератори цієї групи можуть запускатися від ключа або навіть мати систему віддаленого запуску за допомогою пульта дистанційного керування. Кращі бензинові генератори оснащуються системою автоматичного електронного запуску, коли при відключенні енергії агрегат починає роботу самостійно. Такі генератори дуже зручні для резервного живлення підприємств, житлових будинків, інших великих і критично важливих об'єктів. Ціни на такі генератори досить високі.

1.6. Призначення бензинових електростанцій та автоматики

Раптові перебої енергопостачання - ситуація неприємна і, на жаль, типова практично для всіх власників дач і приватних будинків, офісів та магазинів. Якщо ж це відбувається часто, то крім незручностей може завдавати і великі збитки. В лабораторіях і медичних установах тривала перерва в роботі чутливої апаратури може стати згубним. На підприємствах зупинка робочого процесу може призвести до великої кількості браку. Джерела безперебійного живлення та акумуляторні батареї малоефективні при частих і тривалих відключеннях. Для вирішення даної проблеми допоможе тільки установка незалежної побутової електростанції - бензоелектричного (або іншого типу) генератора.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

Однак зручність та безпека в експлуатації бензогенератора значно зросте в разі установки на нього системи автоматичного управління. Адже в разі зникнення електрики система сама запустить генератор (з урахуванням температури двигуна, довкілля і особливостей кожної конкретної моделі) і перемкне споживачів з мережі на генератор, а також зробить коректне відключення генератора при появі напруги в мережі. Бензинові генератори зі стабілізатором напруги АВР або інверторні генератори виробляють струм високої якості. Напруга на виході має мінімальне відхилення, тому вони є ідеальним вибором, як для приватного будинку, так і для бізнесу.

Існує електроніка, для якої застосування генераторів зі стабілізатором не тільки рекомендовано, а й необхідно. Це чутлива медична апаратура в клініках, надточне обладнання в лабораторіях, наукових дослідницьких центрах, різна оргтехніка в офісах: факси, модеми, комп'ютери. Якщо в будинку є що-небудь з перерахованих приладів, а також ноутбуки або мобільні телефони, які потребують постійної підзарядки, то доцільно купити електростанцію з АВР, або інверторний генератор. Вони відмінно захистять найпоширенішу техніку, яка негативно реагує на скачки напруги. Дуже чутливі до відхилень напруги холодильники і морозилки, причому, якщо, наприклад, телевізор захищений плавким запобіжником, то холодильник такого захисту не має.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

Якщо власник будинку, де є газовий котел, планує покупку резервної електростанції, то слід придбати інверторний генератор або бензиновий генератор з АВР, це відмінний вибір для живлення електроніки газового котла. Справа в тому, що газові котли, особливо імпортного виробництва, дуже вибагливі до якості струму і до величини напруги. Електронний пусковий блок відразу плавиться всередині, якщо відчує хоча б невеликий стрибок напруги. Якщо пускова електроніка газового котла має деякий захист від перепадів напруги, то вона не вийде з ладу, а просто відключиться, і не буде працювати. І перший, і другий випадок чреваті тим, що котел не працює, а, значить, і будинок не опалюється. Взимку, в морози, ця ситуація може стати критичною.

1.7. Автоматика до генераторів

АВР (автоматичне введення резерву) - це автоматичний перемикач з основної мережі на електростанцію і назад. Генератори з АВР це єдиний правильний вибір, для того щоб уберегти будь-яку електроніку від поломок. Технологія АВР утримує вихідну напругу генератора на постійній величині, не допускаючи відхилень і стрибків. Нормальне відхилення при задіюванні АВР становить до 5% (210-230В). Автоматичний ввід резерву - це блок, який не тільки контролює показники величини напруги, а й стабілізує вихідну напругу через постійну регулювання струму в обмотці. Якісні і дорогі бензинові генератори і дизельні електростанції від провідних світових компаній завжди володіють вбудованою функцією АВР. Стабілізатор напруги відмінно згладжує і усуває зміни в резервної мережі. Це зберігає електричну техніку від поломок. Дана система захищає не тільки підключені прилади, вона ще здатна захистити і сам генератор, дозволяючи йому уникнути перевантажень.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

ATS (Automatic Transfer Switch) - це блок обладнання, в яку може входити АВР, а також це маркування електростанцій, які оснащені блоком автоматики, на панелі генератора має знаходитися спеціальний вихід для підключення зовнішнього модуля (універсальний, та іноді спеціальний роз'єм). Не зазначається що генератор за маркуванням ATS має автоматику. Блок обладнання складається з двох модулів. Один з них має вигляд електричного щита, який здійснює керуючу роль, він встановлюється окремо. Другий, знаходиться всередині бензинового генератора, він несе виконавчу функцію.

ATS складається з:

1. Електричний блок складається з мікропроцесора з заданим програмним забезпеченням. Від нього відходять з'єднувальні кабелі (контакторна група), які перемикають навантаження від генератора до мережі або назад. Контролер здійснює постійний контроль і діагностику генераторної установки і електричної мережі. Від роботи залежить автоматичний запуск станції і її зупинка. Електронний блок перемикає навантаження від одного контактора на інший. На ньому також може відображатися поточна інформація про робочі характеристики. Та має з'явиться повідомлення про аварійну ситуацію, якщо вона виникне.

Виконавчий блок необхідний для виконання команд мікроконтролера. Він управляє повітряною заслінкою двигуна генератора і іншими його вузлами, щоб здійснити запуск або зупинку.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

На рис. 1.2. зображені елементи генератору які можуть бути підключені до автоматики (ATS).

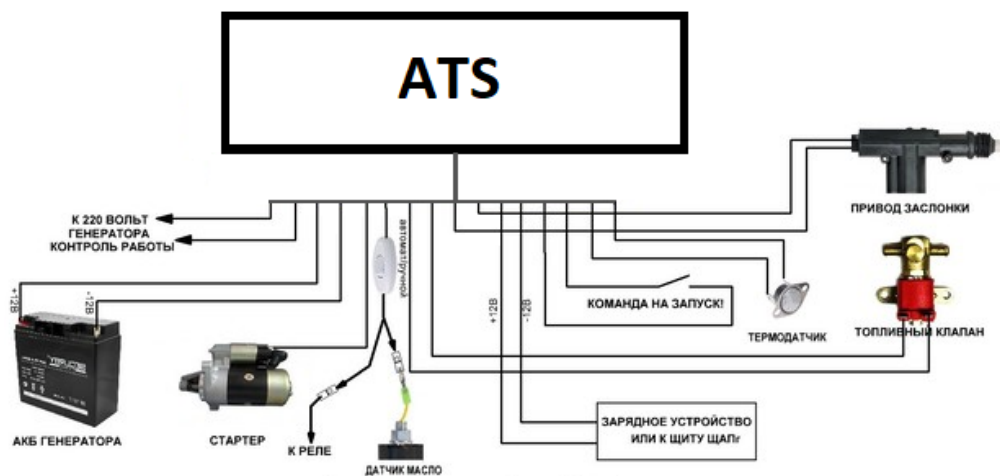


Рис. 1.2. Елементи генератору підключені до автоматики (ATS).

Система автоматичного введення резерву (АВР) та автоматика ATS передбачає контроль напруги і запуск бензогенератора в роботу при відключенні централізованого електропостачання. Контролер, який знаходиться у виконавчому блоку, здійснює комутацію споживача до бензогенераторів і назад до мережі після відновлення електропостачання за допомогою системи контакторів з взаємним блокуванням. Блокування потрібно для уникнення випадків, коли споживач одночасно підключений і до мережі, і до бензогенераторів. Крім автоматичного включення бензогенератора для об'єкта в роботу і його контролю система здійснює підзарядку вбудованого в електростанцію акумулятора, щоб той завжди був готовий до прокручування стартера.

Система АВР є майже в усіх бензогенераторах але ATS може поставлятися як в комплекті з бензогенератором, так і окремо від нього.

ВИСНОВКИ ДО РОЗДІЛУ 1

Аналіз огляду генераторів і автоматики до них показав:

1. Побутові агрегати номінальною потужністю до 10 кВт здебільшого не мають автоматики (ATS), але більшість має систему автоматичний ввід резерву (ABR).
2. Без електростартеру на генераторі автоматику (ATS) неможливо застосувати.
3. Знайдені можливості покращання автоматики, тому що деякі системи не мають всіх функцій інших, а ті що мають коштують занадто дорого.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

РОЗДІЛ 2

ВИМОГИ ДО РОЗРОБЛЯЄМОГО КОНТРОЛЕРУ

Визначимо вимоги до виконавчого блоку:

- Інтелектуальне управління запуском агрегату в залежності від його температури і інших параметрів;
- Індикацію параметрів, режимів роботи і аварійних ситуацій з їх повної розшифровкою на РКІ;
- Продуманий інтерфейс російською мовою;
- Гнучке налаштування всіх основних параметрів;
- Функції енергозбереження (0.3 Вт - в режимі простою);
- Контроль наробки агрегату;
- Можливість оновлення програмного забезпечення;
- Дистанційне керування виконавчим блоком.

Контролер повинен забезпечувати виконання наступних функцій:

- Запуск генератора в автоматичному режимі роботи при зниженні напруги нижче допустимого рівня $U_c < 175V$ (80%);
- Управління заслінкою (ЕПЗ) під час запуску генератора при температурі двигуна нижче заданої (T_g);
- Прогрівання двигуна, якщо його температура нижче заданої (T_n);
- Автоматичний перехід живлення навантаження на мережу і останів генератора в автоматичному режимі роботи при відновленні напруги мережі;
- Охолодження двигуна на холостому ходу;
- Прийом навантаження в разі зникнення напруги мережі при охолодженні;
- Напіваавтоматичний запуск генератора з відключенням мережі і включенням живлення навантаження від генератора;
- Напіваавтоматичний останів генератора з відключенням живлення навантаження від генератора і включенням мережі;

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		20

- Захист лінії генератора від перевантаження і короткого замикання;
- Захист навантаження від високої напруги мережі $U_c > 264V$ (120%);
- Екстрений останов генератора без охолодження якщо натиснута кнопка аварійної зупинки;
- Індикацію основних параметрів: заряду акумулятора, напруги мережі, температури і напруцювання агрегату;
- Видачу аварійних попереджень:
 - при відсутності запуску агрегату;
 - при розряді акумуляторної батареї;
 - при перевищенні допустимого напруги мережі;
 - при відсутності струму, що виробляється генератором (при його роботі).

Помітка: контролер допускає установку в виробничому або житловому (офісному) приміщенні і підключається до генератора, силової мережі та споживачам додатковим монтажем. Блок реле, привід заслінки (ЕПЗ) і датчик температури встановлюються безпосередньо на корпусі бензоелектричного агрегату. Датчик температури кріпиться під гвинт, із застосуванням гровера, безпосередньо на двигун агрегату. Полярність підключення датчика повинна бути дотримана.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

2.1. Вимоги до програмного забезпечення

Для роботи контролера програмне забезпечення не потребується. Для можливості оновлення програмного забезпечення необхідно наступне:

- Atmel Studio 7; [6]
- Windows XP/7/Vista/8/10, або MacOS, або Linux;
- USB 2.0+.

Для розробки плати - програма PCAD 2006 SP2 [7].

2.2. Вимоги до технічної характеристики плати

В таблиці 2.1. зазначені технічні вимоги до плати.

Таблиця 2.1.

Технічні вимоги до плати.

№	Параметр	Значення
1	Електроживлення	(160-230В)AC; (8-16В)DC
2	Пікова споживана потужність	5 Вт
3	Середня споживана потужність в режимі роботи	< 1 Вт
4	Максимальна потужність комутації контактів управління	250В, 8А (AC-1)
5	Діапазон робочих температур	Від -10 до 50°C(-40°C)
6	Габаритні розміри / Вага (ШхГхВ)	110х90х68мм /~0,3кг
7	Діапазон вимірюваних температур	-25 - +90 °C крок 1°C
8	Діапазон вимірюваної напруги акумуляторної батареї	0 - 20 В крок 0,1В

На рис. 2.1. зображена блок схема контролера.

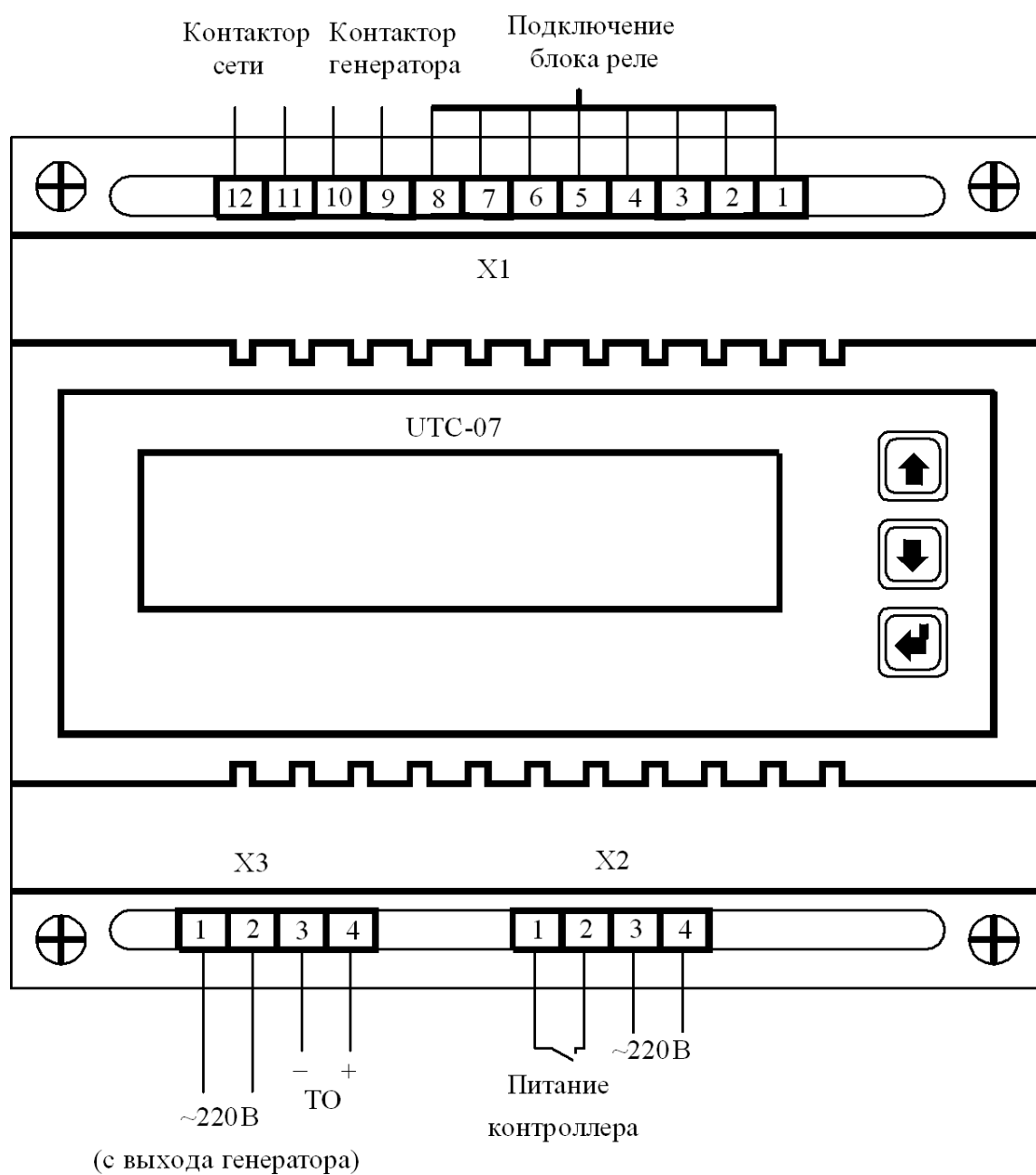


Рис. 2.1. Блок схема контролеру.

ВИСНОВКИ ДО РОЗДІЛУ 2

Аналіз вимог до розробляемого контролеру, до програмного забезпечення, до технічної характеристики плати, а також аналіз огляду агрегатів (див. “ВИСНОВКИ ДО РОЗДІЛУ 1”) визначає наступне:

1. Виконавчий блок буде мати обмежену пам'ять, тому треба ретельно її розподіляти. Для цього обрана мова програмування C. [8]
2. Код буде розділений на 2 частини: на графічну частину для відображення на РКІ екрану, та бізнес логіку контролера. За принципом MOVE (Model, Objects, View, Events).
3. Дистанційне керування буде розроблено як окрема система, описана у розділі “СИСТЕМИ ДЛЯ ДИСТАНЦІЙНОГО КЕРУВАННЯ”.
4. Для розробки плати обрана програма PCAD 2006 SP2.
5. Для того щоб зумовити можливість оновлення програмного забезпечення, а саме створення прошивки, обрана програма Atmel Studio 7.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

РОЗДІЛ 3

ПРОЕКТУВАННЯ ТА РОЗРОБКА КОНТРОЛЕРУ

3.1. Заходи безпеки

Монтажні та пусконаладжувальні роботи повинні виконувати організації або особи, які мають необхідну кваліфікацію. Перед введенням автоматизованого генератора в експлуатацію необхідно вивчити будову та правила експлуатації генератора і блоку автоматичного керування. Перед підготовкою виробу до включення необхідно вивчити даний технічний опис, ознайомитися з конструкцією блоку, функціональними схемами і алгоритмами роботи системи. Перевірити вірність з'єднання згідно зі схемою підключення. Перевірити, чи правильно підключено напруги мережі до виробу. При експлуатації пристрою необхідно керуватися чинними правилами техніки безпеки при експлуатації електроустановок і додатково виконувати викладені нижче заходи безпеки. Не зберігати поблизу генератора горючі і легкозаймисті матеріали. Не вмикати генератор в роботу без його заземлення. Перед включенням автоматики переконатися в правильності підключення акумуляторної батареї (дотримання полярності). Чи не торкатися під час роботи виробу до струмоведучих частин, що знаходяться під напругою, не підключати і не відключати кабелі при наявності напруги на відповідних роз'ємах і клеммах. Не проводити ремонтні роботи і не усувати несправності генератора і автоматики під напругою. При ремонті і обслуговуванні генератора всі роботи виконувати після вимкнення живлення автоматики щоб уникнути запуску двигуна. Перед зняттям з генератора або установкою акумуляторної батареї вимкнути живлення запропонованого контролеру і відключити зарядний пристрій (при його наявності).

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

3.2. Розміщення і монтаж системи

Система з контролером розміщується безпосередньо в приміщенні поряд з генератором, в будь-якому місці зручному для експлуатації та обслуговування. Контролер може бути встановлений в існуючий щиток на стандартну DIN-рейку.

Підключення зовнішніх електричних ланцюгів до блоку ATS здійснюється за допомогою гермовводи зверху чи знизу пристрою згідно зі схемою підключення.

З'єднання блоку ATS з блоком реле здійснюється 8-жильним кабелем. При цьому використовується кабель будь-якого типу, перетин проводів - від 0.35мм². При значній відстані генератора від блоку управління (20м ~ 30м) необхідно використовувати кабель з перетином від 1,5 мм².

Силова частина ATS з контролером підключається між розподільним щитком з автоматичними вимикачами або запобіжниками і лічильником електроенергії. Повинен бути передбачений автоматичний або інший вимикач для відключення системи від мережі для монтажу, ремонту та інших робіт. Вимикач (запобіжник) може встановлюватися до лічильника або після нього.

Принцип підключення для мережі з заземленою нейтраллю показаний на рис. 3.1., з ізольованою нейтраллю - на рис. 3.2, та на рис. 3.3.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

Якщо сумарне навантаження споживачів перевищує потужність агрегату, через АВР підключаються тільки споживачі, які потребують безперервного енергопостачання, це зображено на рис. 3.4.

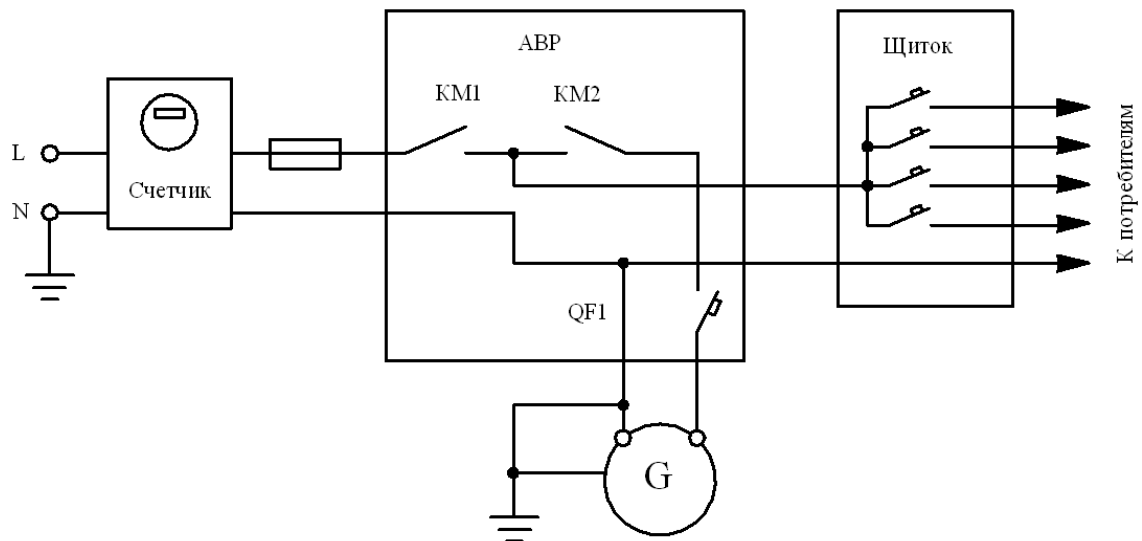


Рис. 3.1. Однофазна мережа з заземленою нейтраллю.

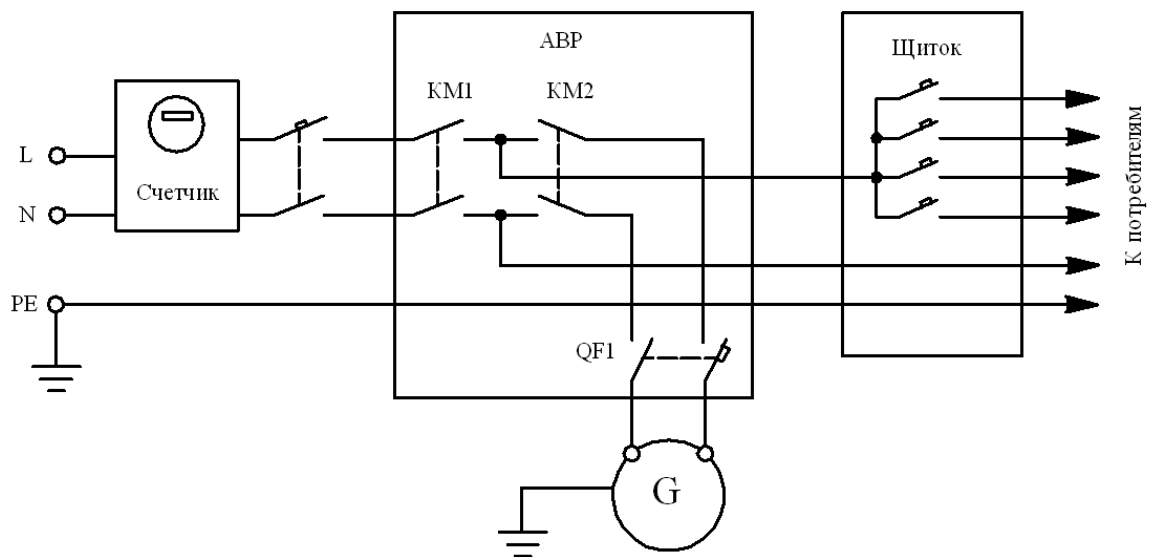


Рис. 3.2. Однофазна мережа з ізольованою нейтраллю.

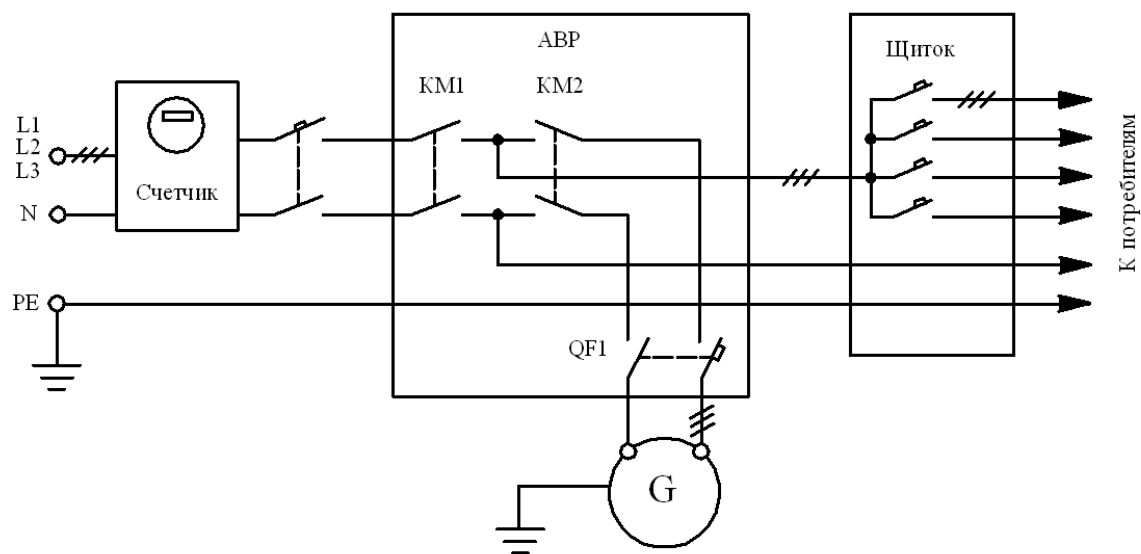


Рис. 3.3. Трифазна мережа з ізолюваною нейтраллю.

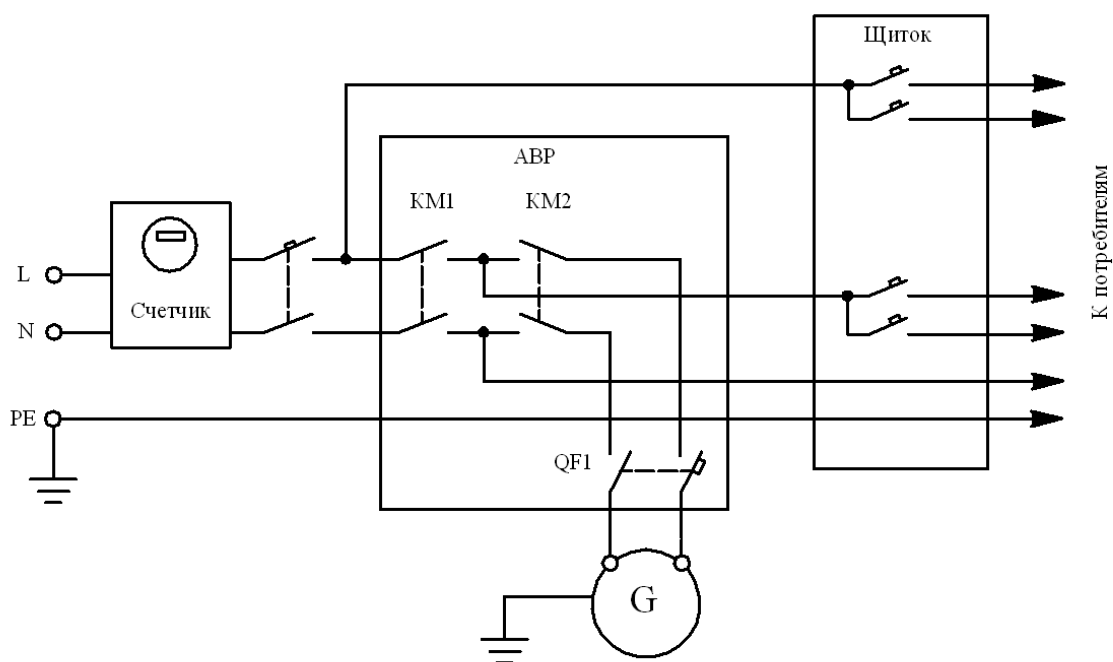


Рис. 3.4. Однофазна мережа з ізолюваною нейтраллю, сумарне навантаження перевищує потужність генератора.

3.3. Конструкція системи

Електричний блок, частина автоматики (ATS), змонтований в металевому боксі зі ступенем захисту IP54, зображений на рис. 3.5.

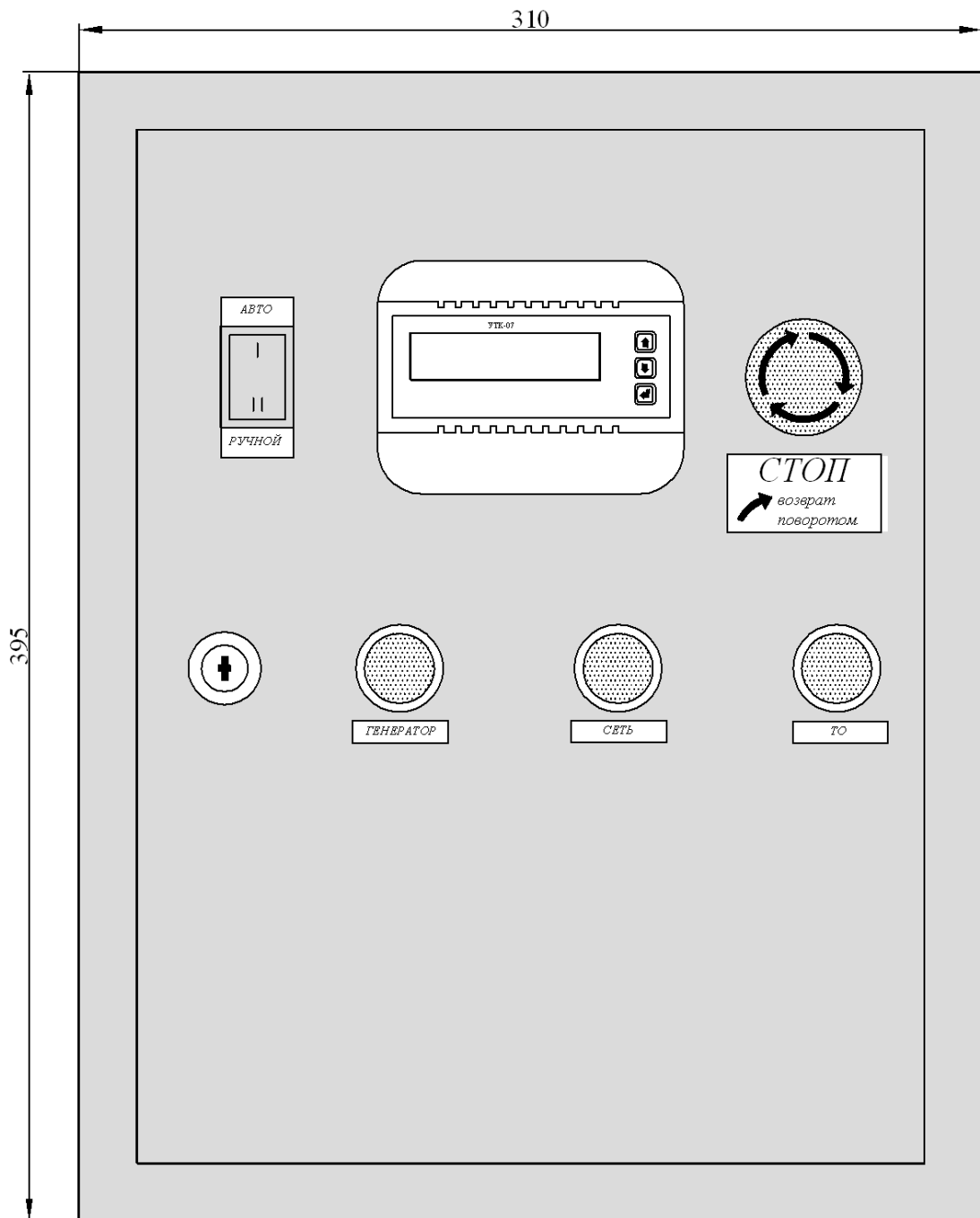


Рис. 3.5. Зовнішній вигляд електричного блоку.

На лицьовій панелі боксу розташовані:

- тумблер перемикання режиму роботи пристрою (Авто / Контактор генератора);
- кнопка аварійної зупинки, яка також використовується для включення / виключення живлення автоматики (кнопка червоного кольору з поворотним поверненням);
- лампочка зеленого кольору «Мережа» (горить при наявності напруги мережі);
- лампочка зеленого кольору «Генератор» (горить після перемикання навантаження на генератор);
- лампочка червоного кольору «ТО» (загоряється і блимає, при необхідності виконати технічне обслуговування генератора, перший раз через 20 годин роботи генератора, далі через кожні 100 годин.).

Всередині боксу встановлений безпосередньо блок управління контролером, а також силова частина, що складається з одного трифазного автоматичного вимикача і двох контакторів.

Електричний блок для зручності експлуатації і додаткового захисту, виконаний в окремому корпусі з кріпленням на DIN рейку. На передній панелі блоку знаходиться двухстрочний рідкокристалічний дисплей, на який виводиться вся інформація про роботу пристрою. Також на передній панелі розташовані кнопки управління блоком.

Для зручності користування блок кріпиться в боксі під напівпрозорою поворотною кришкою розташованої на лицьовій панелі боксу.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		30

До складу системи автоматики входить виконавчий блок (блок-реле), який встановлюється безпосередньо на генераторі, зображений на рис. 3.6.



Рис. 3.6. Виконавчий блок.

3.4. Шаблони програмування, що використовувались

MOVE - “Models, Operations, Views, Events”, патерн програмування, для поділу даних програми. В інших шаблонах та патернах програмування часто виникає проблема в тому, що контролери мають занадто багато коду та логіки. [9]

На рис. 3.7. зображена діаграма, яка показує базову структуру MOVE.

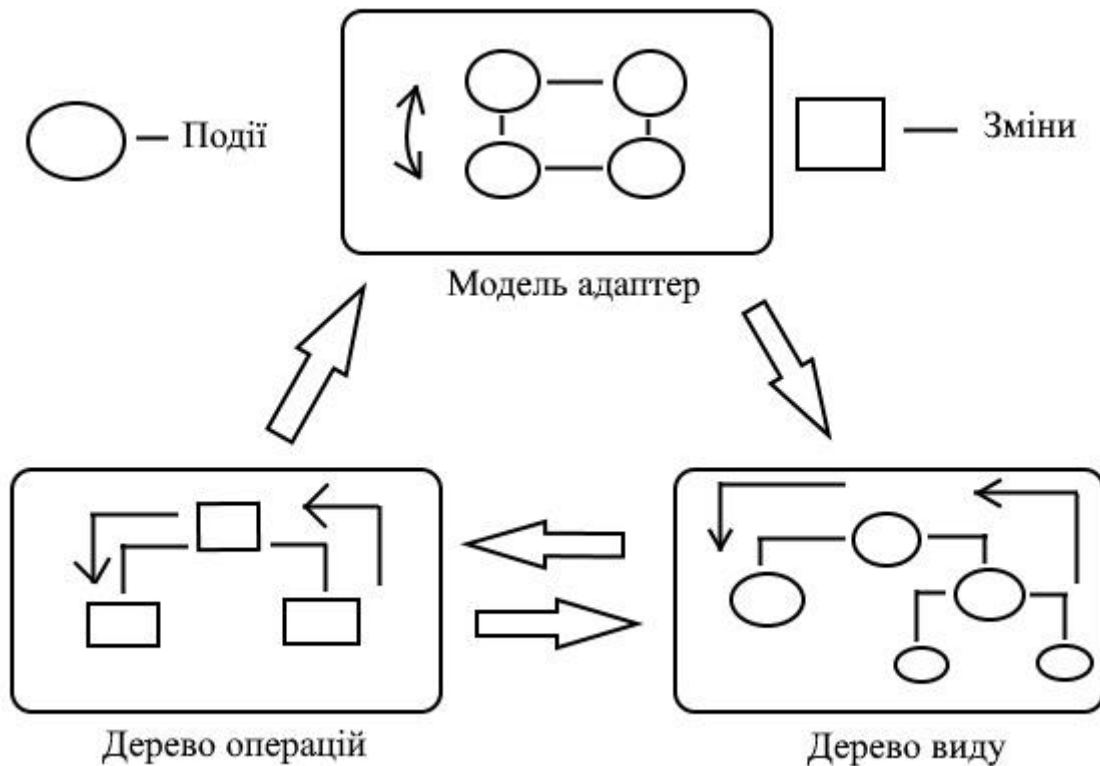


Рис. 3.7. діаграма базової структури MOVE.

Модель Адаптер (Model Adapter) - надає дані і реагує на команди контролера, змінюючи свій стан. У програмі з використанням MOVE, моделі тільки у виді обгортки даних. Це означає що вони можуть містити функції, що дозволяють перевіряти, наприклад пароль, але не містять функції, які зберігають дані в базу або завантажують їх у стороннє API. Це робота «операцій».

Операції (Operations) - це все що програма виконує. Вони несуть відповідальність за зміни у вашій моделі, показують правильні уявлення в потрібний час, і реагують на події запуснені взаємодією користувача. В добре побудованій програмі, кожна суб-операція може бути запущена незалежно від її батька; саме тому на діаграмі див. рис. 3.7. події йдуть по вихідній, а зміни по низхідній. Використання операцій таким чином, дозволяє розглядати сам додаток як дію, яка починається при старті програми. Вона породжує стільки суб-операцій скільки потрібно, де одночасно існуючі суб-операції запускаються паралельно, вихід з програми відбувається після завершення всіх подій.

Вид (Views) - посередник між програмою і користувачем, відповідає за відображення даних моделі користувачеві, реагуючи на зміни моделі. Це уявлення, яке відповідає за показ форм взаємодії для користувача. Коли користувач взаємодіє з програмою, уявлення генерує подію, яка містить введені користувачем дані. Все, що користувач може бачити або з чим взаємодіяти, має бути реалізовано в уявленні. Воно не тільки відображає стан прикладної програми в зрозумілій формі, а й також є потоком взаємодії з користувачем в зрозумілій події. Важливим є те, що уявлення не змінює нічого в моделі безпосередньо, воно просто запускає події з операціями, і чекають змін, слухаючи події, що запускаються моделями.

Події (Events) використовуються щоб безпечно з'єднувати всі компоненти та інтерпретує дії користувача, сповіщаючи модель про необхідність змін. Подія запускає уявлення, коли користувач взаємодіє. Крім того, коли операція завершилася, модель запускає події, щоб повідомити вашому додатку, що вона змінилася.

3.5. Режими роботи

Пристрій має чотири режими роботи:


- ІНФО;
- ПОЛУАВТОМАТ;
- АВТОМАТ;
- РУЧНИЙ.

ІНФО - інформаційний режим. В цей режим контролер переходить при включенні живлення. При цьому відображаються параметри: температура двигуна генератора T , напруга акумулятора U , час напрацювання генератора N і напруга мережі живлення U_c .

ПОЛУАВТОМАТ - запуск і останов генератора відбувається примусово, за командами від кнопки на панелі блоку автоматики. У цьому режимі, перехід живлення навантаження на генератор після запуску відбувається навіть при наявності напруги. Основне призначення напівавтоматичного режиму - завчасний перехід на живлення від генератора при необхідності відключення лінії мережі або при відомому часу відключення електропостачання.

АВТОМАТ - автоматичний режим роботи забезпечує енергопостачання без участі обслуговуючого персоналу. Вибирається за допомогою кнопки на панелі блоку ATS. При пропажі напруги здійснюється запуск генератора і видача напруги генератора на навантаження. Якщо з будь-якої причини не відбувся запуск генератора, на дисплей виводиться напис: "АВАРИЯ ГЕНЕРАТОР НЕ ЗАПУСТИЛСЯ". При відновленні напруги мережі, автомат відключає навантаження від генератора і переключає на мережу та зупиняє генератор.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

Для режимів «АВТОМАТ» і «ПОЛУАВТОМАТ» необхідно тумблери на блоці реле і на блоці автоматики перевести в положення «АВТО». Якщо з якої-небудь причини в роботі пристрою стався збій, і генератор зупиняється, на дисплей блоку виводиться повідомлення «Авария» та розшифровка її причини. Контролер може бути переведений з цього стану в режим «ІНФО», утриманням клавіші . Для повторного запуску необхідно усунути причину виникнення аварії (можливо - змінити налаштування контролера) і провести повторний запуск.

РУЧНИЙ - запуск може бути здійснено і при відключеному блоці автоматики. Для запуску генератора необхідно:

- Вимкнути живлення автоматики (натиснути червону кнопку аварійної зупинки на передній панелі пристрою). Перемкнути тумблер блоку автоматики в положення «КОНТАКТОР ГЕНЕРАТОРА». (Ця дія примусово перемкне контактор генератора на навантаження, розімкнувши мережевий контактор);
- Перемкнути тумблер «АВТО / РУЧНИЙ» на блоці реле, в положення «РУЧНИЙ»;
- Запустити генератор ключем.









3.6. Налаштування параметрів

Блок автоматики має ряд параметрів, що настраюються забезпечують коректну роботу пристрою:

- t_1 - час після якого, відбудеться запуск генератора при зникненні основного електроживлення, а також повернення на мережу при поновленні електроживлення. Діапазон регулювання 2 - 10с;
- t_2 - час роботи стартера генератора при включенні запалювання. Діапазон регулювання 1 - 6с;
- t_3 - час роботи генератора з заслінкою (прогрів). Діапазон регулювання 5 - 30с;
- Тип приводу ("ЗАСЛОНКА / КЛАПАН") - Параметр встановлюється в залежності від типу агрегату;
- t_4 - час охолодження генератора після відключення навантаження. Діапазон регулювання 5 - 120с;
- t_5 - час прогріву генератора до підключення навантаження. Діапазон регулювання 5 - 180с;
- T_g - температура, нижче якої прогрів генератора здійснюється із закритою заслінкою. Діапазон регулювання 0 - 60°C;
- T_n - температура, при якій підключення навантаження відбувається без прогріву. Діапазон регулювання 0 - 60°C;
- TO - скидання сигналу TO.

Помітка: реальний час роботи стартера обмежена блокуванням, яка виникає при виході генератора на задані обороти. Таким чином, виконується захист стартера і продовження його терміну служби.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

Для переходу в режим налаштування параметрів після включення блоку в мережу необхідно натиснути і тривало утримувати кнопку . При цьому на дисплеї відображається поточне значення параметра, змінити яке можна повторно натиснувши кнопку , при цьому значення параметра починає блимати. Далі за допомогою кнопок   потрібно виставити потрібне значення даного параметра. Зберегти виставлене значення можна натисканням кнопки . Пересування між параметрами здійснюється також за допомогою кнопок  . Для виходу з режиму налаштування параметрів необхідно вибрати пункт ВИХІД, далі натиснути і утримувати кнопку .

3.7. Алгоритм работы исполнительного блока

На рис. 3.8. Зображена структура меню.

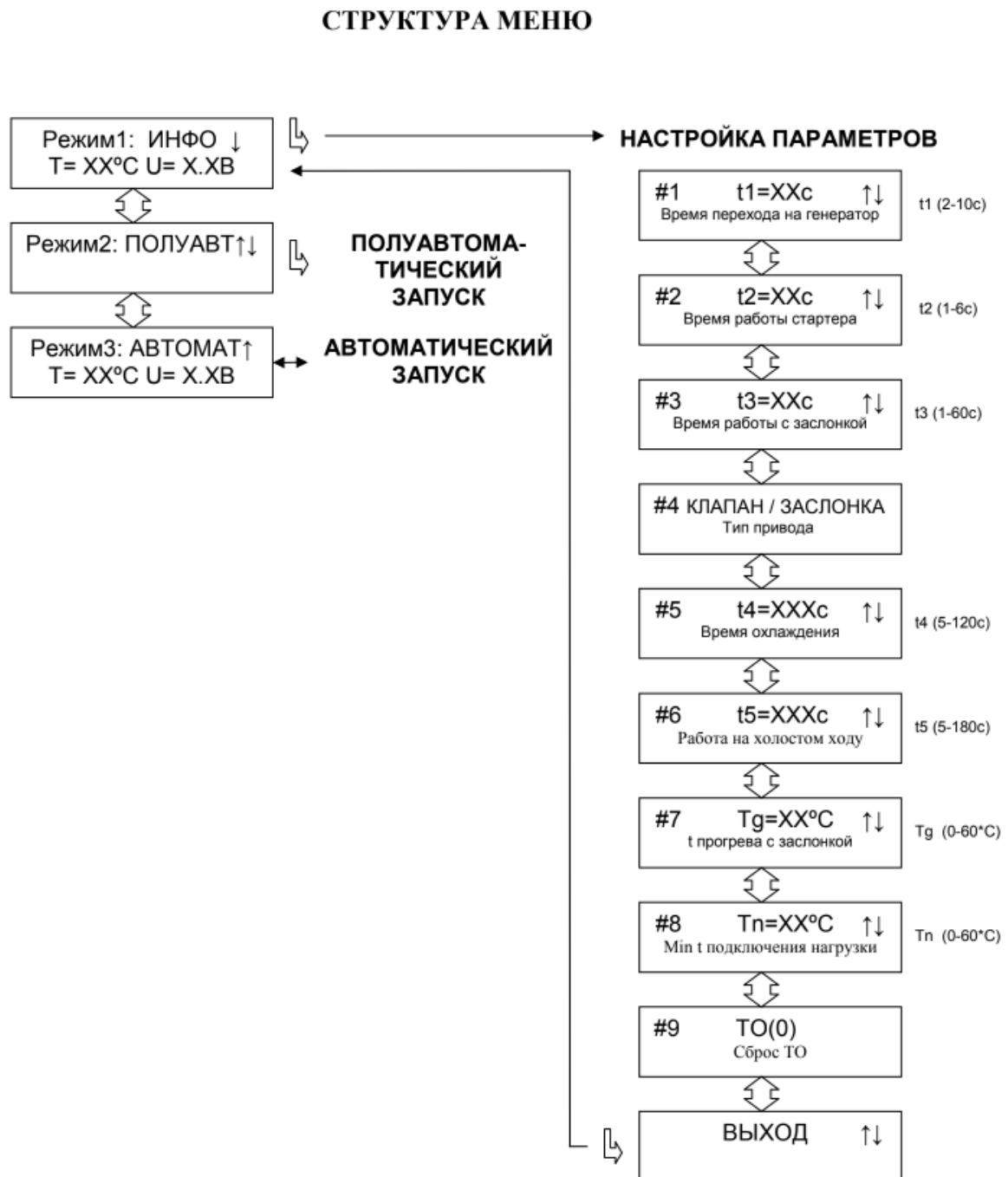


Рис. 3.8. Структура меню.

При зниженні напруги мережі нижче допустимої межі, в режимі “АВТОМАТ”, контролер очікує час t_1 . Після чого, якщо мережа не з'явилася, а напруга акумуляторної батареї в межах норми - починається запуск. Контролер перевіряє температуру корпусу генератора, при цьому поточна температура T порівнюється із заданою температурою двигуна генератора T_g .

У разі, коли $T > T_g$ відбувається включення стартера і подача запалювання (час роботи стартера визначається параметром t_2), в разі, коли температура менше заданої, перед включенням стартера відбувається закриття повітряної заслінки (ЕПЗ) і тільки потім прогрів генератора (час роботи з заслінкою задано параметром t_3).

У деяких типах агрегатів використовується ЕМ клапан замість повітряної заслінки. Режим роботи вибирається параметром «Тип приводу». На наступному етапі виконується розгін (час розгону - 5С). У разі якщо з якоїсь причини двигун зупинився, вичікують пауза 5 секунд і вся процедура спроби запуску повторюється. Перед останньою спробою пауза збільшена в два рази (10с), щоб акумулятор встиг відновитися. Кількість спроб запуску - три. Якщо після третьої спроби генератор так і не запустився, вся автоматика переводиться в режим останів та на дисплей виводиться повідомлення про аварію.

Якщо генератор запустився успішно, відбувається перевірка поточної температури генератора і порівняння її з T_n - заданої мінімальної температурою, при якій може бути підключена навантаження. Якщо температура генератора більше T_n - виконується підключення навантаження. У разі, коли $t < T_n$, виконується прогрів генератора на холостому ходу протягом заданого часу (параметр t_5) і лише потім підключається навантаження.

Перед підключенням навантаження виконується перевірка напруги лінії генератора. Якщо напруга генератора низька або не з'явилося - виводиться повідомлення про аварію.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

Робота автоматики при підключеному навантаженні зводиться до постійної перевірки відсутності мережі. При появі основного електроживлення, навантаження перемикається на нього лише через певний період часу (t_1).

На рис. 3.9. Зображена блок-схема підключення навантаження, де:
 t_1 - час переходу з мережі на генератор чи навики.

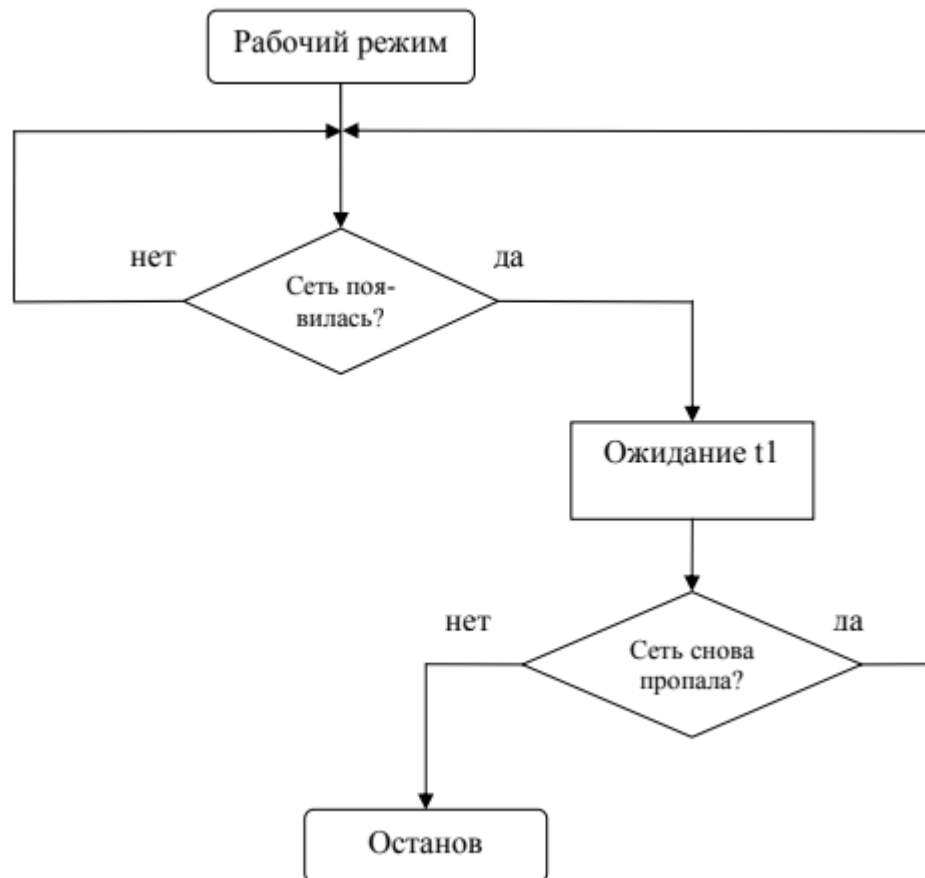


Рис. 3.9. Блок-схема підключення навантаження.

На рис. 3.10. зображена блок-схема “Запуск”, де:

t - поточна температура генератора;

T_g - температура генератора, нижче якої відбувається прогрів з заслінкою;

T_n - мінімальна температура підключення навантаження;

K_z - кількість спроб запуску генератора ($K_z = 4$).

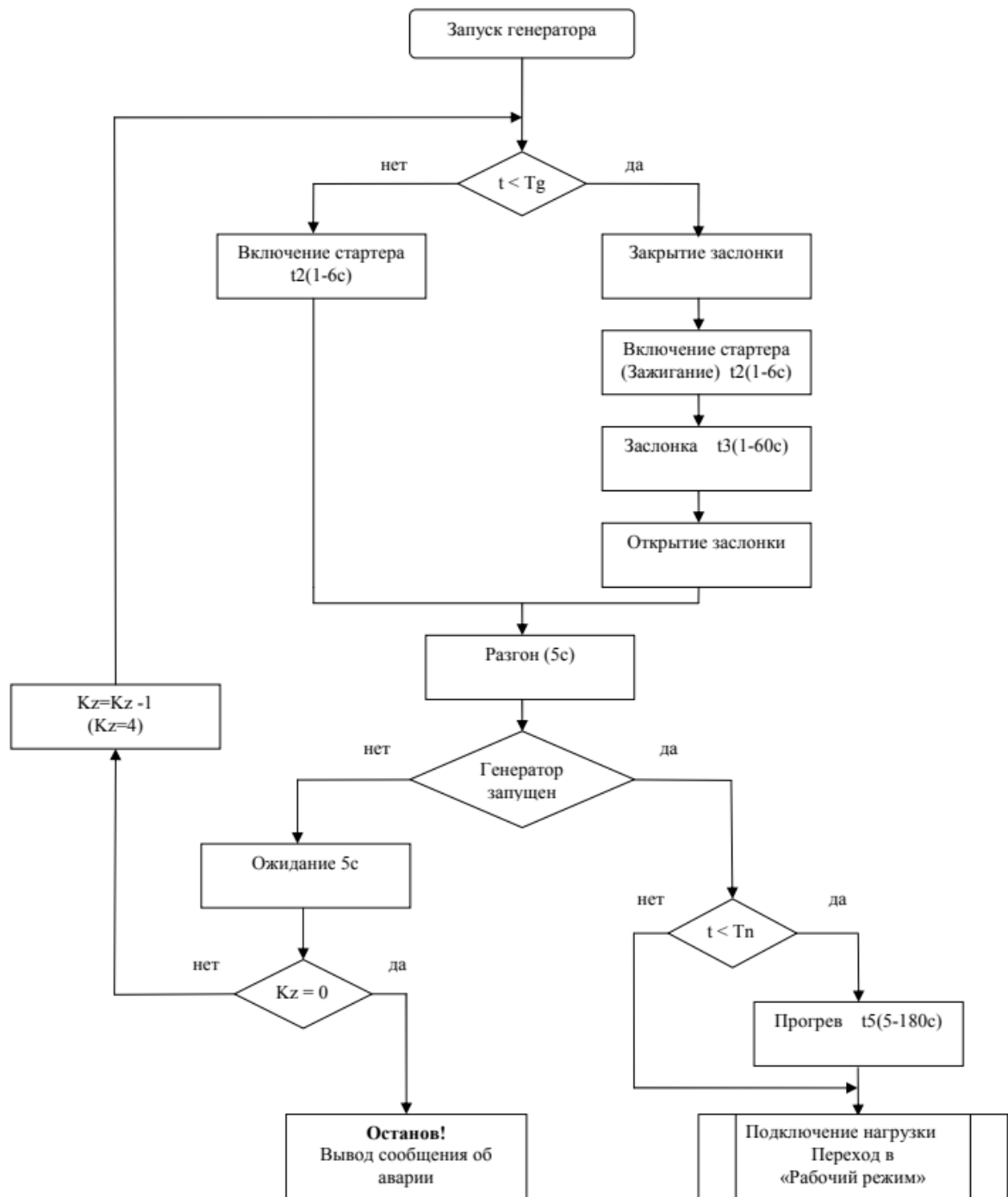


Рис. 3.10. Блок-схема "Запуск".

Нормальний останов генератора виконується наступним чином: після відключення навантаження від генератора вичікують пауза 1 секунда (час перемикання контакторів), навантаження підключається до мережі і виконується охолодження генератора протягом заданого часу (параметр t_4). Якщо в «останов» ми перейшли з режиму «АВТОМАТ», то при повторному зникненні мережі генератор, протягом часу t_4 знову перейде в режим «РОБОТА». У разі перевищення нормального напруги мережі (220В) більше 20% відбувається моментальне відключення контакторів і останов генератора в будь-якому з режимів. Після чого контролер видає аварійний повідомлення. На рис. 3.11. зображена блок-схема "Останов".

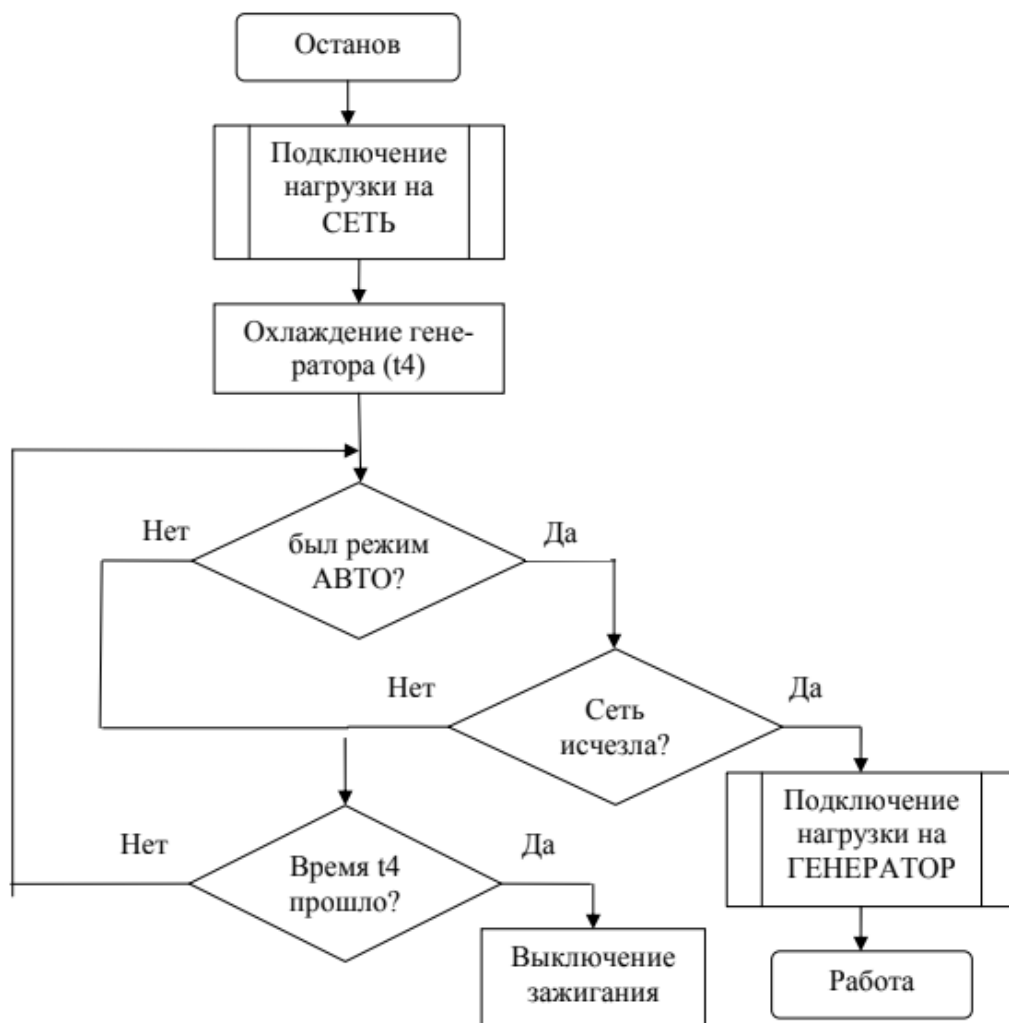



Рис. 3.11. Блок-схема “Останов”.

3.8. Аварійна зупинка генератора

Примусовий останов генератора може бути виконаний на будь-якому етапі запуску, роботи або нормального зупинки в автоматичному чи напівавтоматичному режимі. Для цього необхідно утримувати клавішу , або натиснути кнопку аварійної зупинки, на передній панелі пристрою. При цьому зупинка двигуна відбудеться без охолодження.

3.9. Рекомендовані параметри

В таблиці 3.1. зазначені рекомендовані параметри налаштування.

Таблиця 3.1.

Рекомендовані параметри налаштування.

Параметр		Значення
t1	Час переходу на генератор	6
t2	Час роботи стартера	3
t3	Час роботи з заслінкою	4
t4	Час охолодження	10
t5	Час холостого ходу	20
Tg	T прогріву з заслінкою	10
Tn	Min T підключення навантаження	30

3.10. Повідомлення про помилки

В таблиці 3.2. наведені повідомлення про помилки.

Таблиця 3.2.

Повідомлення про помилки.

Повідомлення (Російською мовою)	Причина
ГЕНЕРАТОР НЕ ЗАПУСТИЛСЯ	<ul style="list-style-type: none">Виникає, якщо всі спроби запуску пройшли невдало. Запуск визначається станом лінії «контроль ротора».<ol style="list-style-type: none">виник обрив лінії «контроль ротора»;неправильно налаштовані параметри;виникла неполадка генератора;в баку відсутній бензин;перемикач «автомат / ручний» на блоці управління або на блоці реле знаходиться не в тій позиції
АККУМУЛЯТОР РАЗРЯЖЕН	<ul style="list-style-type: none">При спробі запуску заряд акумуляторної батареї нижче порогового рівня (10В):<ol style="list-style-type: none">аккумулятор вимагає зарядки;аккумулятор відсутній;блок реле генератора не підключений до контролера, або лінія обірвана;

Таблиця 3.2. продовження

НЕТ ТОКА ОТ ГЕНЕРАТОРА	<ul style="list-style-type: none"> Виникає, при спробі переведення навантаження на генератор, якщо він працює, але на силовому виході немає напруги. Або якщо в режимі «РОБОТА» зникла напруга: <ol style="list-style-type: none"> відключений вступної автомат генератора; виникла неполадка генератора; в баку закінчився бензин.
НАПРЯЖЕНИЕ СЕТИ ПРЕВЫШЕНО >264В	<ul style="list-style-type: none"> Виникає, при появі в лінії «мережу» небезпечного для навантаження напруги. Необхідно перевірити напругу лінії мережу, і усунути причину виникнення неполадки; !!! Не вимикайте контролер в цьому режимі до перевірки і усунення причини підвищеної напруги. Інакше висока напруга може бути подано в навантаження.

3.11. Створення прошивки

Для створення прошивки використовується програма Atmel Studio 7. Створимо новий проект за допомогою команди меню File -> New -> Project, як зображено на рис. 3.12.

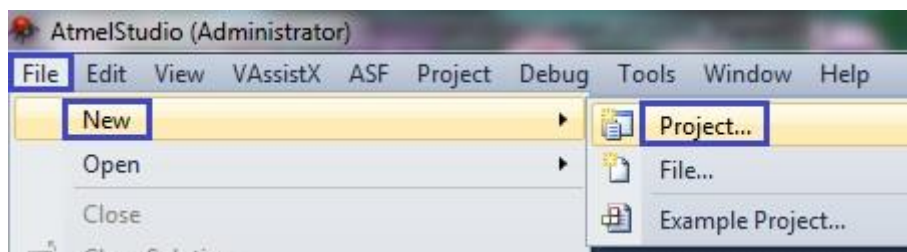


Рис. 3.12. Новий проект в Atmel Studio.

У нас відкриється діалог, в якому буде 5 варіантів проектів на C / C ++, один вид проекту на асемблері, а також є варіант створення порожнього рішення, в яке вже потім можна додавати проекти. Ми будемо писати саме на мові C. За допомогою кнопки "browse" в даному діалозі ми вибираємо папку, в якій будемо створювати свої проекти, виберемо варіант проекту "GCC C Executable Project", дамо проекту ім'я, наприклад "Test01". Галочка зліва від напису "Create directory for solution" означає, що буде створюватися папка для проекту, тобто дану папку заздалегідь створювати не потрібно. Натиснемо "ОК". Це зображено на рис. 3.13.

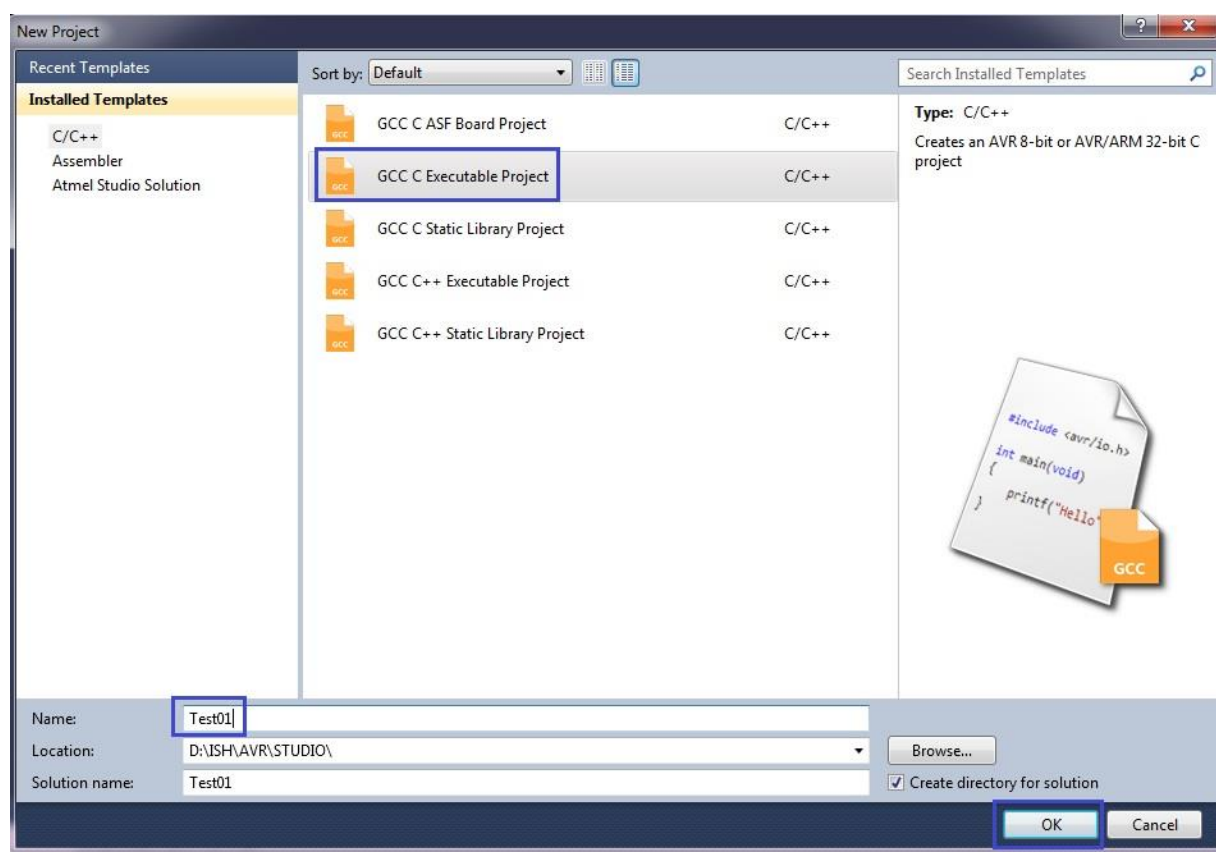


Рис. 3.13. Вибір варіанту проекту в Atmel Studio.

У наступному діалозі виберемо наш мікроконтролер Atmega8A, впечатав це у відповідному віконці. Справа також є “Datasheet” - посилання на технічну документацію для нашого мікроконтролера, яку ми звідси можемо завантажити. Натискаємо "ОК". Це зображено на рис. 3.14.

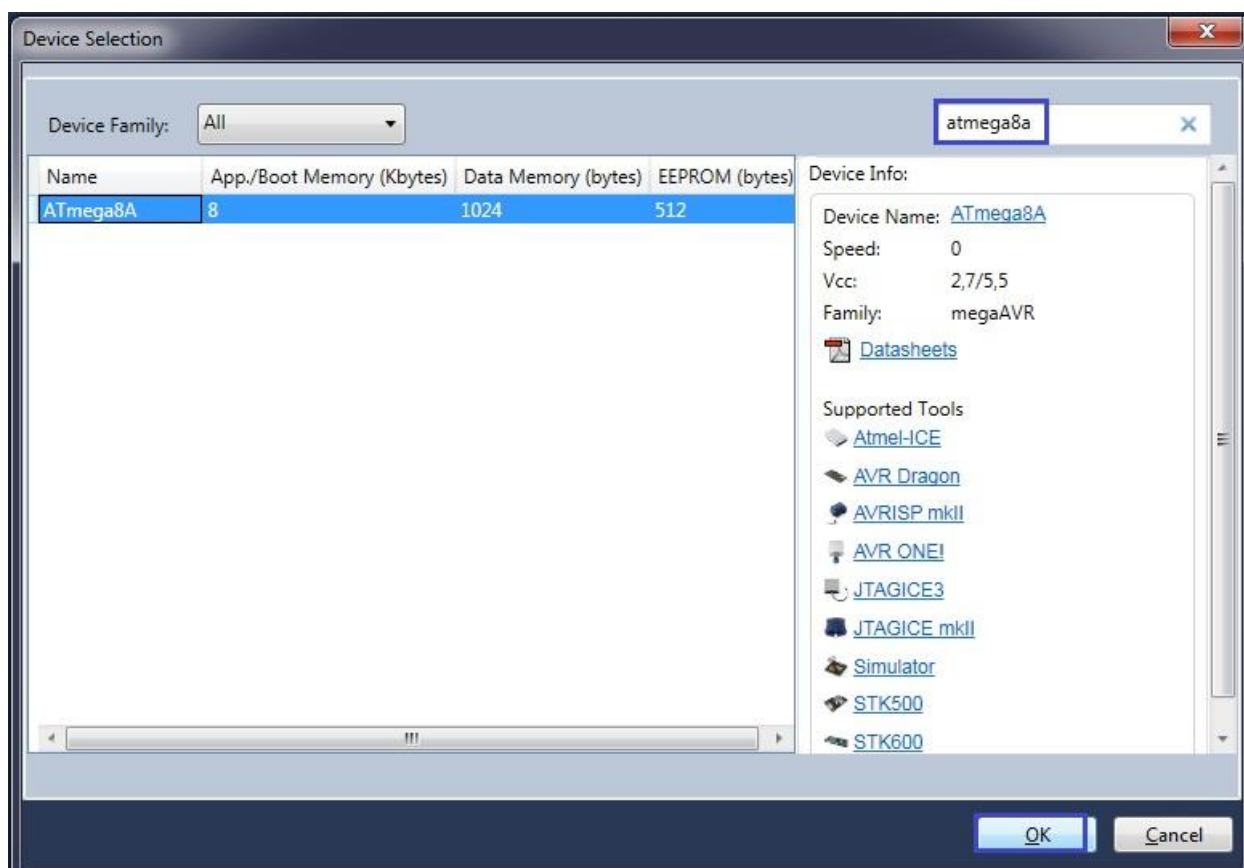


Рис. 3.14. Вибір контролера в Atmel Studio.

Проект створений. Ми повинні побачити вікно з редактором коду. Створюємо код, або копіюємо. Потім спробуємо скомпілювати наш проект (або "зберемо", так як проект даним процесом не тільки компілюється, але ще і лінкуються). Натискаємо відповідну кнопку в панелі інструментів, або натискаємо функціональну клавішу "F7". Після цієї дії, якщо збирач проекту не зустрине ніяких помилок, то ми отримаємо в нижній частині нашого середовища програмування певний повідомлення, зображене на рис. 3.15.

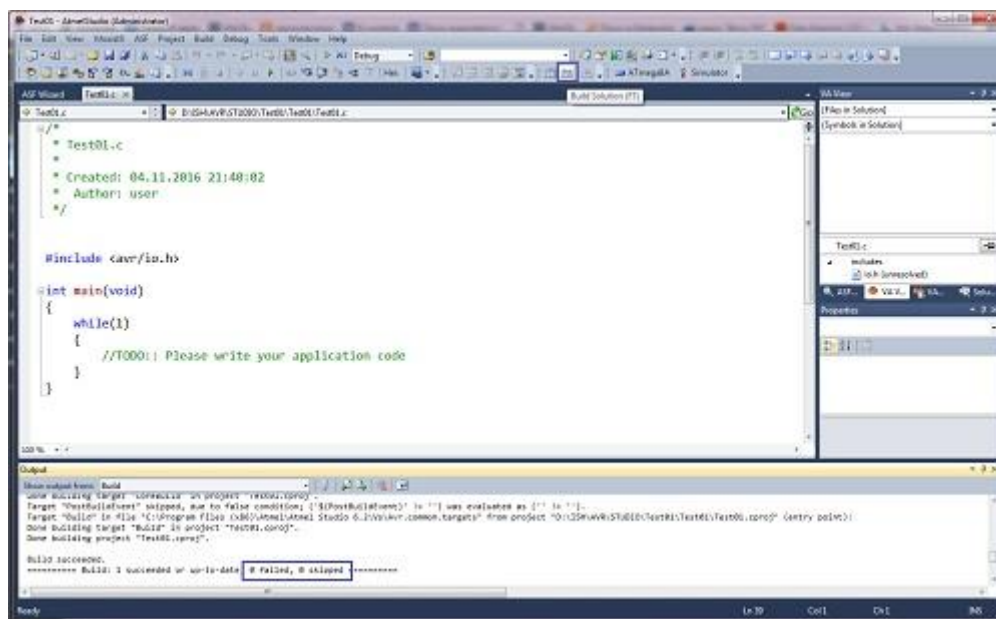


Рис. 3.15. Компілювання проекту в Atmel Studio.

В папці нашого проекту у нас буде згенеровано виконавчий файл, та файл прошивки, з розширенням ".hex". Даний файл ми і будемо "прошивати" на наш контролер через USB.

Примітка: ці дії можна повторити для коду для РКІ екрану, код для контролера і для екрана незалежні один від одного.

ВИСНОВКИ ДО РОЗДІЛУ 3

1. Програма Atmel Studio дуже допомагає у створенні прошивки, спрощує процес, та допомагає виявити помилки, які можуть виникнути, саме до обраного контролера (у нашому випадку - Atmega8A).
2. Після тестування на декількох генераторах, виявилось що створити універсальний контролер неможливо, бо деякі генератори використовують стару версію датчиків, приводу заслінки, та інше, тому для них потрібна інша версія прошивки контролера.
3. Даний розділ може бути використано у навчальному процесі в практичному курсі.
4. В якості мови програмування була обрана мова Objective C, бо пам'ять контролеру дуже обмежена і потребує підвищеного контролю.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		50

РОЗДІЛ 4

СИСТЕМИ ДЛЯ ДИСТАНЦІЙНОГО КЕРУВАННЯ ГЕНЕРАТОРОМ ЧЕРЕЗ GSM/GPRS

GSM (від назви групи Groupe Spécial Mobile, пізніше перейменований в Global System for Mobile Communications) (рос. СПС-900) - глобальний стандарт цифрового мобільного стільникового зв'язку з розділенням каналів за часом (TDMA) і частоті (FDMA).

GPRS (англ. General Packet Radio Service - «пакетний радіозв'язок загального користування») - надбудова над технологією мобільного зв'язку GSM, що здійснює пакетну передачу даних. GSM забезпечує підтримку таких послуг:

- Послуги передачі даних (синхронний і асинхронний обмін даними, в тому числі пакетна передача даних - GPRS). Дані послуги не гарантують сумісність термінальних пристроїв і забезпечують тільки передачу інформації до них і від них;
- Передача мовної інформації;
- Передача коротких повідомлень (SMS);
- Передача факсимільних повідомлень.

4.1. Опис роботи системи

При отриманні смс пристрій GSM / GPRS Shield, на базі Arduino, буде виконувати одну із зазначених в смс функцій, після чого повідомить про це у зворотній смс. Після подачі живлення треба дочекатися отримання смс з повідомленням про те, що ініціалізація пройшла успішно і модем GSM зареєструвався в мережі. Після цього пристрій готовий до роботи. Пристрій може отримувати повідомлення, що містять параметри "Дія", "Пристрій". Для управління пристроєм необхідно сформувавти смс за наступним алгоритмом:

- Основні функції:
 - [ДІЯ] [ПРИСТРІЙ];
 - Додаткові функції:
 - [ЗНАЧЕННЯ%] - задаємо значення ШІМ-сигналу. [10]

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		51

Після правильно написаної і відправленої інформації дочекайтеся зворотного повідомлення з повідомленням про те, що ваша команда пройшла.

4.2. Перелік необхідного

- GSM/GPRS Shield — дозволяє працювати в мережах стільникового зв'язку за технологіями GSM / GPRS для прийому / передачі даних, відправлення / отримання SMS і здійснення телефонного голосового зв'язку; [11]
- Arduino Uno R3 - основний контролер для підключення модулів; [12]
- Trema Shield — це плата розширення, яка спрощує процес підключення модулів до Arduino. Використання Trema Shield позбавляє від необхідності пайки проводів (наприклад живлення) при підключенні декількох модулів до Arduino, спрощує процес створення;
- Реле; Trema-модуль Твердотельное реле (SSR - Solid-state relay) — призначено для замикання вихідний ланцюга змінного струму (з номінальною напругою до 240В), при подачі керуючого сигналу 5В постійного струму на вхід модуля; [13]
- Battery Shield — це джерело автономного живлення для 5В плат; [14]
- 1х Джерело живлення 12В / 2А;
- 1х Вентилятор (кулер) 12В;
- 1х Силовий ключ - необхідний для управління навантаженням. [15]

Для реалізації проекту необхідно встановити бібліотеки:

- iarduino_GSM - для роботи з GSM / GPRS Shield; [16]
- Бібліотека SoftwareSerial входить в базовий набір Arduino IDE і не вимагає установки.

4.3. Алгоритм роботи системи

Прийом вхідних повідомлень складається з наступних дій:

1. Зчитування повідомлення з пам'яті SIM-карти, після чого функція автоматично їх звідти видаляє.
2. Після зчитування повідомлення запускається цикл, в якому текст повідомлення перевіряється на наявність в ньому керуючих команд.
3. Якщо команда знайдена, то всередині нього запускається другий цикл, в якому текст також перевіряється, але вже на наявність в ньому імені пристрою.
4. Якщо ім'я пристрою знайдено, то виконується перевірка на наявність додаткового параметра, зазначеного з символом "%" - відсоток (від 0 до 100) швидкості, з яким буде працювати вказаний пристрій.
5. Якщо додатковий параметр було вказано, то ставиться флаг про його наявності в отриманому смс.

Виконання зазначених в повідомленні дій:

- Після того, як всі флаги були виставлені, відбувається перевірка наявності флагу додаткового параметра. Якщо він присутній, тоді виконується обчислення значення числа, зазначеного перед символом%:
 - Для цього ми знаходимо кількість символів в рядку, розташованих зліва від символу %;
 - Отримане число покаже нам порядковий номер символу % в масиві strSMStxt;
 - Для того, щоб дізнатися, яке значення було зазначено, перевіriamo 3 найближчі осередки масиву, розташовані зліва від символу %:
 - Якщо 1 комірка, розташована найближче до символу %, містить число від 0 до 9, тоді ми заносимо її значення в змінну Sum. Це буде розряд "одиниці";

- Після цього всередині першої перевірки виконується друга перевірка 2 комірки на наявність числа від 0 до 9. Якщо число знайдено, то воно множиться на 10 і додається до змінної Sum. Це буде розряд "десятки";
- Після цього всередині другої перевірки проводиться третя перевірка 3 комірки. Якщо в ній знайдена 1, то вона множиться на 100 і додається до змінної Sum. Це буде розряд "сотні";
- Якщо ж у третій перевірці виявиться, що в "сотнях" зазначено більше 1, то флаг flgPWM буде скинутий;
- Виконується перевірка наявності флагу flgFunc:
 - Якщо флаг flgFunc був встановлений, тоді виконується перевірка наявності флагу flgDev:
 - Якщо флаг flgDev був встановлений, тоді виконується перевірка наявності флагу flgPWM:
 - Якщо флаг flgPWM встановлений, тоді виконується подача ШІМ-сигналу на виведення із зазначеним в повідомленні пристроєм і відправка смс з повідомленням про успішне виконання;
 - Якщо флаг flgPWM не встановлено, тоді виконується подача сигналу HIGH на виведення із зазначеним пристроєм і відправка смс з повідомленням про успішне виконання;
 - Якщо флаг flgDev не встановлено, тоді проводиться відправка смс повідомлення про те, що в повідомленні не вказано пристрій;
 - Якщо флаг flgFunc не встановлено, тоді проводиться відправка смс повідомлення про те, що прийшло помилкове повідомлення.

Опис коду:

1. До коду `void setup()` визначаються номери виведення, оголошуються змінні і функції, підключаються бібліотеки, і створюються об'єкти для роботи з ними;
2. У коді `void setup()` конфігуруються вибрані виведення, ініціюється робота GSM/GPRS Shield, виконується цикл очікування готовності GSM/GPRS Shield до роботи (реєстрація в мережі оператора). І після виконання всіх зазначених дій виконується оповіщення про готовність до роботи, шляхом відправки смс на вказаний номер;
3. Код `void loop()` виконується тільки при отриманні смс повідомлення. Сам блок розділений на дві основні частини: прийом вхідних повідомлень і виконання зазначених в повідомленні дій.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		55

4.4. Підготовка та збірка

1. Встановити Battery Shield на Arduino Uno R3. Під час установки Battery Shield повинен бути в вимкненому стані.
2. На Battery Shield встановити GSM/GPRS Shield.
3. На платі GSM/GPRS Shield є слот для SIM-карт і перемикач UART. Встановити SIM-карту в слот, вибрати положення перемикача RX-7/TX-8.
4. На GSM/GPRS Shield встановити Trema Shield.
5. Підключити Реле до виведення “4” на Trema Shield.
6. Підключити до виведення “5” на Trema Shield Силовий ключ, а до нього підключити вентилятор (кулер), пустивши провід живлення вентилятора (червоний) безпосередньо до джерела живлення 12В, а провід GND вентилятора (чорний) в розрив через Реле в Trema Shield.

Перед завантаженням прошивки треба обов'язково поставити в змінної SMSnum номер телефону, куди буде приходити смс-повідомлення про пройдену ідентифікацію. Прошивка коду здійснюється через Arduino Uno R3, порт USB, з використанням Arduino IDE.

4.5. Зберігання та транспортування

Плата контролеру, системи дистанційного керування та РКІ екран допускають зберігання в упаковці в капітальних опалювальних і неопалювальних приміщеннях, що забезпечують збереження виробів від механічних впливів і забруднень з навколишнього середовища, що не містить агресивних парів і газів.

Зберігання має проводитися в наступних умовах:

- температура повітря від -30С до + 60С;
- відносна вологість повітря до 80% при температурі + 25С.

Розроблені системи можуть транспортуватися усіма видами транспорту, з дотриманням правил перевезення вантажів діючих на даному виді транспорту, в пакувальному ящику за умови захисту від прямого впливу атмосферних опадів і пилу. Системи повинені транспортуватися в умовах, що не перевищують заданих граничних умов зберігання. Після транспортування або зберігання при мінусовій температурі, перед установкою необхідно витримати пристрій при плюсовій температурі протягом 12 годин.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

ВИСНОВКИ ДО РОЗДІЛУ 4

1. У період тестування система працювала надійно, без збоїв. Засоби зв'язку працювали стійко і надійно. Однак іноді запит даних з мобільного телефону приходив із затримкою відповіді від 5 хвилин до години, але система спрацьовувала правильно: запуск або останов генератора відбувався лише 1 раз, хоча запитів приходило декілько.
2. Система дистанційного керування збирається легко, що дозволяє використовувати її в якості навчального матеріалу. Але в такому випадку потрібно написати інструкцію, або детальну документацію, бо в цій роботі потрібні знання, яких може не бути в учнів.
3. Дана система вимагає наявності комплектуючих, але ціна на останніх невисока.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

РОЗДІЛ 5

ТЕСТУВАННЯ ВСІХ МОДУЛІВ НА ОБ'ЄКТІ

Розроблені у цій роботі модулі:

1. Контролер генератора.
2. РКІ екран.
3. Системи дистанційного керування.

Склад системи автоматики в таблиці 5.1.

Таблиця 5.1.

Склад системи автоматики.

Найменування	Примітка
Бокс з силовою частиною.	У боксі встановлено 2 контактора, 3-х фазний автомат та повністю виконаний електромонтаж.
Розроблений контролер	Поставляється в пластиковому корпусі з кріпленням на DIN рейку.
Блок реле	Блоку у комплекті з кабелем 8 жив (довжина 15м.).
Датчик температури	ПДС (-50 - +150°C) 10мВ/°К
Привід заслінки	±12В, 1А
Зарядний пристрій	12В, 2А

На рис. 5.1. Зображен бокс з силовою частиною.



Рис. 5.1. Бокс з силовою частиною.

На рис. 5.2. Зображен блок реле.



Рис. 5.2. Блок реле.

На рис. 5.3. Зображена система автоматики “ЩИТОК ГАРАЖ”.



Рис. 5.3. Система автоматики “ЩИТОК ГАРАЖ”.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

На рис. 5.4. Зображен генератор з встановленим блок реле (20м від системи автоматики).



Рис. 5.4. Генератор з встановленим блок реле.

На рис. 5.5. Зображен розроблений контролер з підключеним РКІ екраном

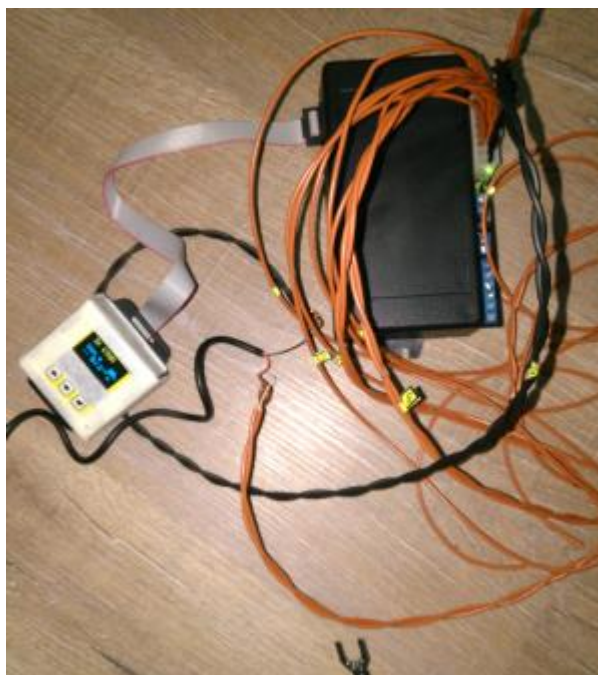


Рис. 5.5. Розроблений контролер з підключеним РКІ екраном.

На рис. 5.6. Зображен розроблений контролер без захисного корпусу.

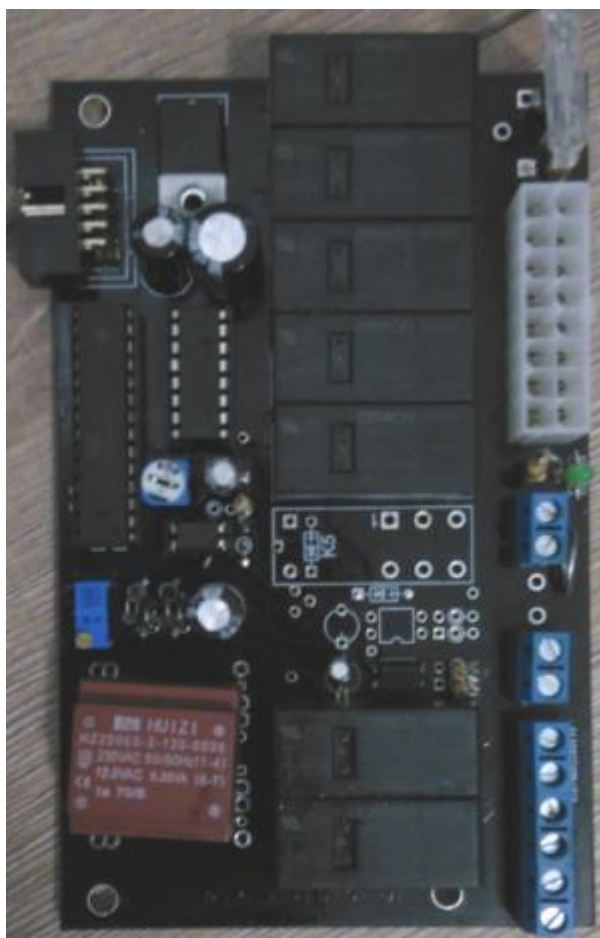


Рис. 5.6. Розроблений контролер без захисного корпусу.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

На рис. 5.7. зображена узагальнена схема підключення всіх модулів, де:

- Розроблений контролер генератора, який є частиною виконавчого блоку (див. “3.3. Конструкція системи”), встановлений у “ЩИТОК ГАРАЖ”;
- РКІ екран вмонтований в бокс з силовою частиною;
- Системи дистанційного керування були розташовані окремо від щитка, але у “ЩИТОК ГАРАЖ”, бо вони повинні бути підключені до електричного блоку;
- “ЕЛЕКТРОСТАНЦІЯ” це бензогенератор (випробування проводилися з генератором Honda EP6500);
- “СТАБІЛІЗАТОР” це АВР, він встановлюється якщо генератор не має функції АВР.

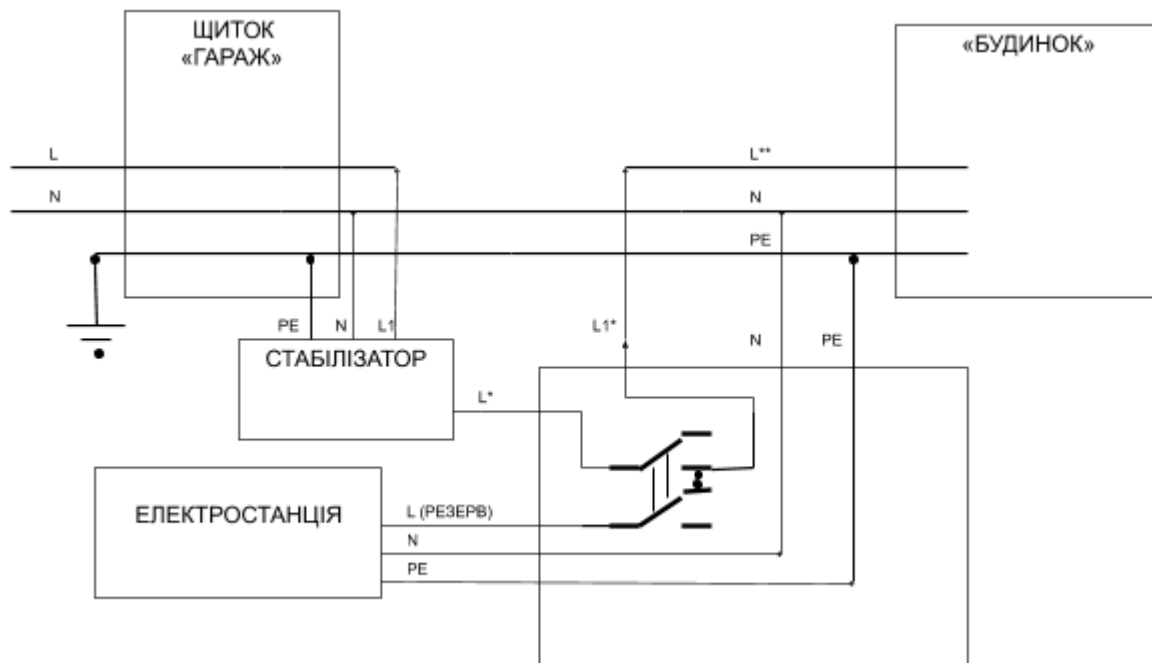


Рис. 5.7. Узагальнена схема підключення всіх модулів.

ВИСНОВКИ ДО РОЗДІЛУ 5

1. Бензогенератори з електростартером підлягають автоматизації з передпідготовки: установка соленоїдного приводу для управління повітряної заслінкою карбюратора і комутація сигнальних проводів замка запалювання і вивід цих провідників для управління від автоматики.
2. Монтажні та пусконаладжувальні роботи повинні виконувати організації або особи, які мають необхідну кваліфікацію. Автоматизація бензинових електростанцій вимагає уважного і професійного підходу, тому що від проведення цих робіт залежить надійність запуску генератора в період відключень зовнішнього електроживлення.
3. Установка контролера не вимагає внесення будь-яких змін що ведуть до втрати гарантії на генератор.
4. Клас даного контролера не вимагає обов'язкової сертифікації.
5. Установка системи на інші моделі генераторів також можлива, але як правило вимагає внесення дрібних змін в код і та вивчення схеми генератора фахівцем.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		65

ВИСНОВКИ

1. У представленій роботі були проаналізовані мікроконтролери, та на основі технічних вимог обран “atmega8”.
2. Був розроблен контролер керування автоматичним запуском побутових електростанцій.
3. Було проведено тестування контролера на декількох генераторах, та аналіз результатів тестування, які показали, що використані методи розробки, усі реалізовані функції дозволяють конкурувати контролеру на ринку, або ця робота може бути навчальним матеріалом.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

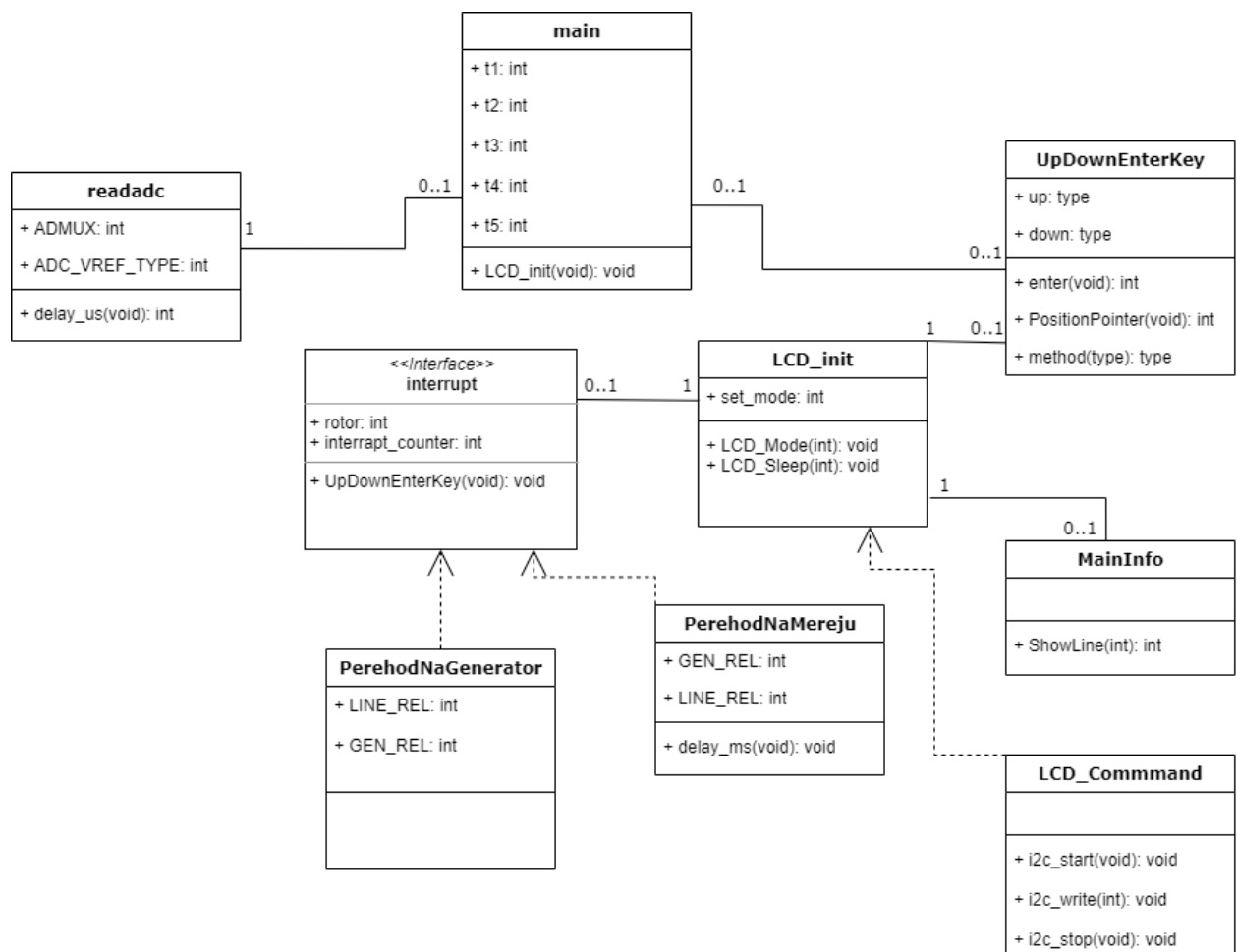
1. “Як влаштовані генератори постійного та змінного струму” - [посилання](#).
2. “Двигуни внутрішнього згоряння. Теорія.”. Підручник / В.Г. Дяченко; За ред. А.П. Марченка. — Харків : НТУ «ХПІ», 2008. — 488 с.
3. Wikipedia Генератор змінного струму - [посилання](#).
4. Синхронні машини, навчальний фільм - [посилання](#).
5. Бурман, А.П.; Строев, В.А. Современная электроэнергетика. В 2 томах. — 4-е, перераб. и доп.. — М.: МЭИ, 2008. — 632 с.
6. Документація Atmel Studio 7 - [посилання](#).
7. “P-CAD 2006. Керівництво схемотехніка, адміністратора бібліотек, конструктора”, автор Саврушев Е. Ц., Рік видання: 2007.
8. Книга “Programming in Objective-C, Sixth Edition”, Автор: Stephen G. Kochan, рік видання: 2014.
9. Патерн “MOVE”, <https://habr.com/en/post/147038/>.
10. ШІМ сигнал на Wiki - [посилання](#).
11. GSM/GPRS Shield - <https://iarduino.ru/shop/Expansion-payments/gsm-gprs-shield.html>.
12. Arduino - <https://iarduino.ru/shop/arduino/arduino-uno-r3.html>.
13. Trema реле - <https://iarduino.ru/shop/rele/rele-trema-modul.html>.
14. Battery Shield - <https://iarduino.ru/shop/Expansion-payments/battery-shield.html>.
15. Trema силовий ключ - <https://iarduino.ru/shop/Expansion-payments/silovoy-klyuch-trema-modul.html>.
16. Бібліотека iarduino_GSM - <https://iarduino.ru/file/345.html>.

Схема електрична структурна
Контролер керування автоматичним запуском
побутових електростанцій

Діаграма методів.

Аркушів 1

2020

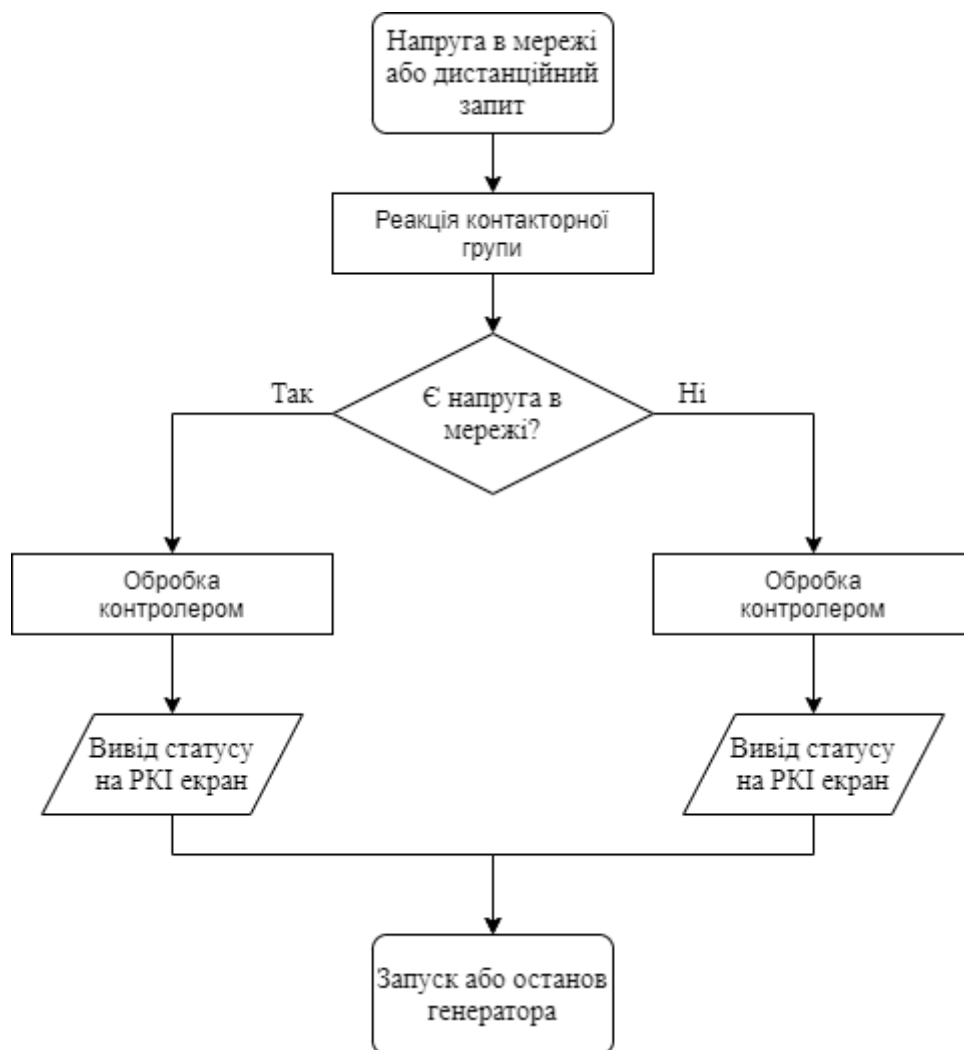


					ІАЛЦ.467100.004 Е1		
Зм.	Арк.	№ докум.	Підпис	Дата	Контролер керування автоматичним запуском побутових електростанцій Діаграма методів.		
Розробив	Баранник Д.Є.						
Перевір.	Таран В. І.						
Н. контр.	Сімоненко В.П.						
Затверд.							
					Лім.	Арк.	Аркушів
						1	1
					КПІ ім. Ігоря Сікорського Каф. ФІОТ, гр. ІО-361		

Схема електрична функціональна
Контролер керування автоматичним запуском
побутових електростанцій

Блок-схема алгоритму взаємодії.

Аркушів 1



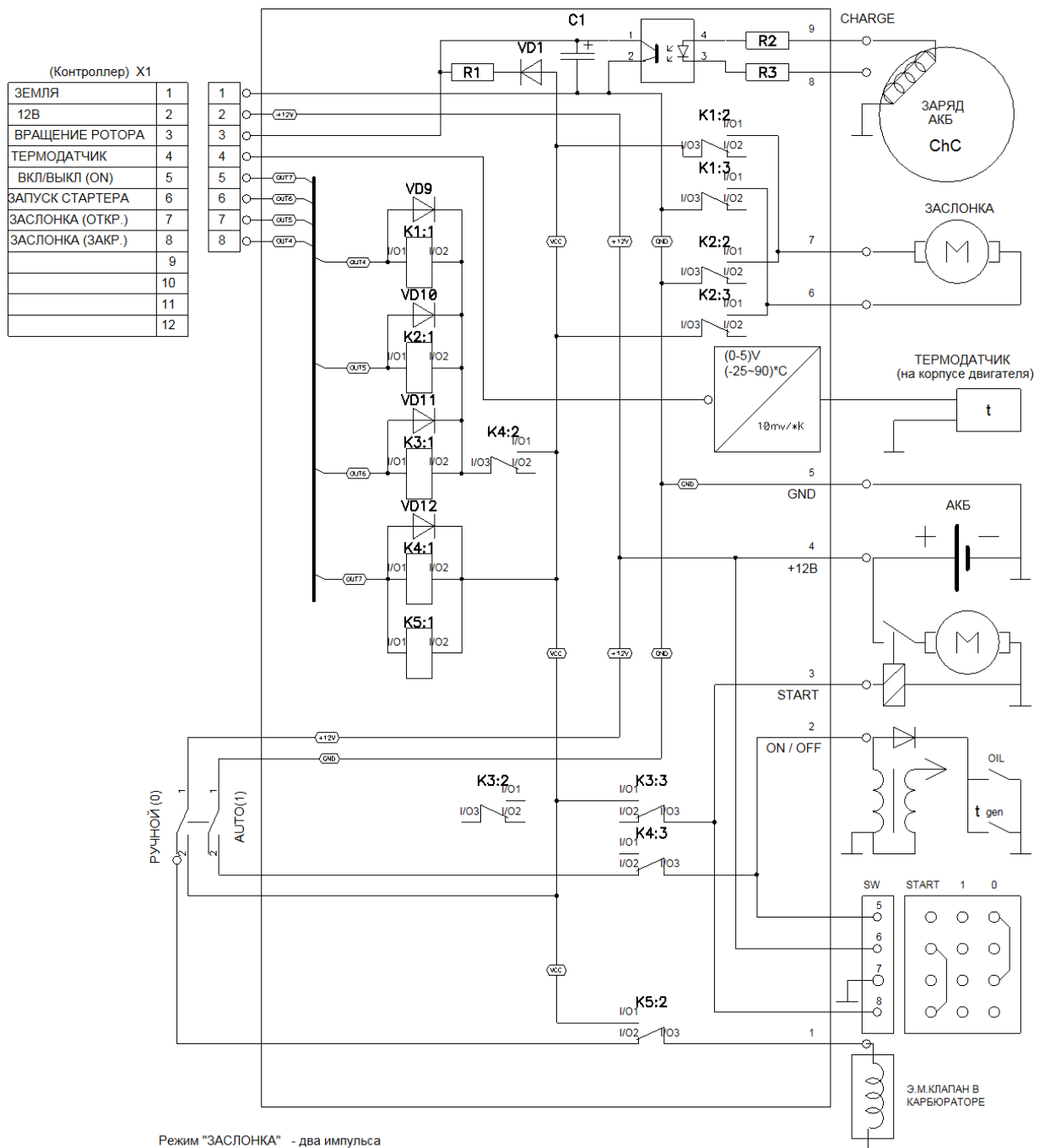
					ІАЛЦ.467100.005 Е2		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розробив		Баранник Д.Є.			Контролер керування автоматичним запуском побутових електростанцій Блок-схема алгоритму взаємодії.	Літ.	Арк.
Перевір.		Таран В. І.					1
							1
Н. контр.		Сімоненко В.П.				КПІ ім. Ігоря Сікорського Каф. ФІОТ, гр. ІО-361	
Затверд.							

Схема електрична принципова
Контролер керування автоматичним запуском
побутових електростанцій

Схема підключення систем до бензоелектричного агрегату.

Аркушів 1

2020



РЕЖИМ "ЗАСЛОНКА" - два импульса
противоположной полярности на ВЫВОДЫ 7 и 6
(закрытие и открытие заслонки)

РЕЖИМ "КЛАПАН" - один продолжительный импульс
(+12В на ВЫВОД 7, Земля на ВЫВОД 6)

ІАЛЦ.467100.006 ЕЗ

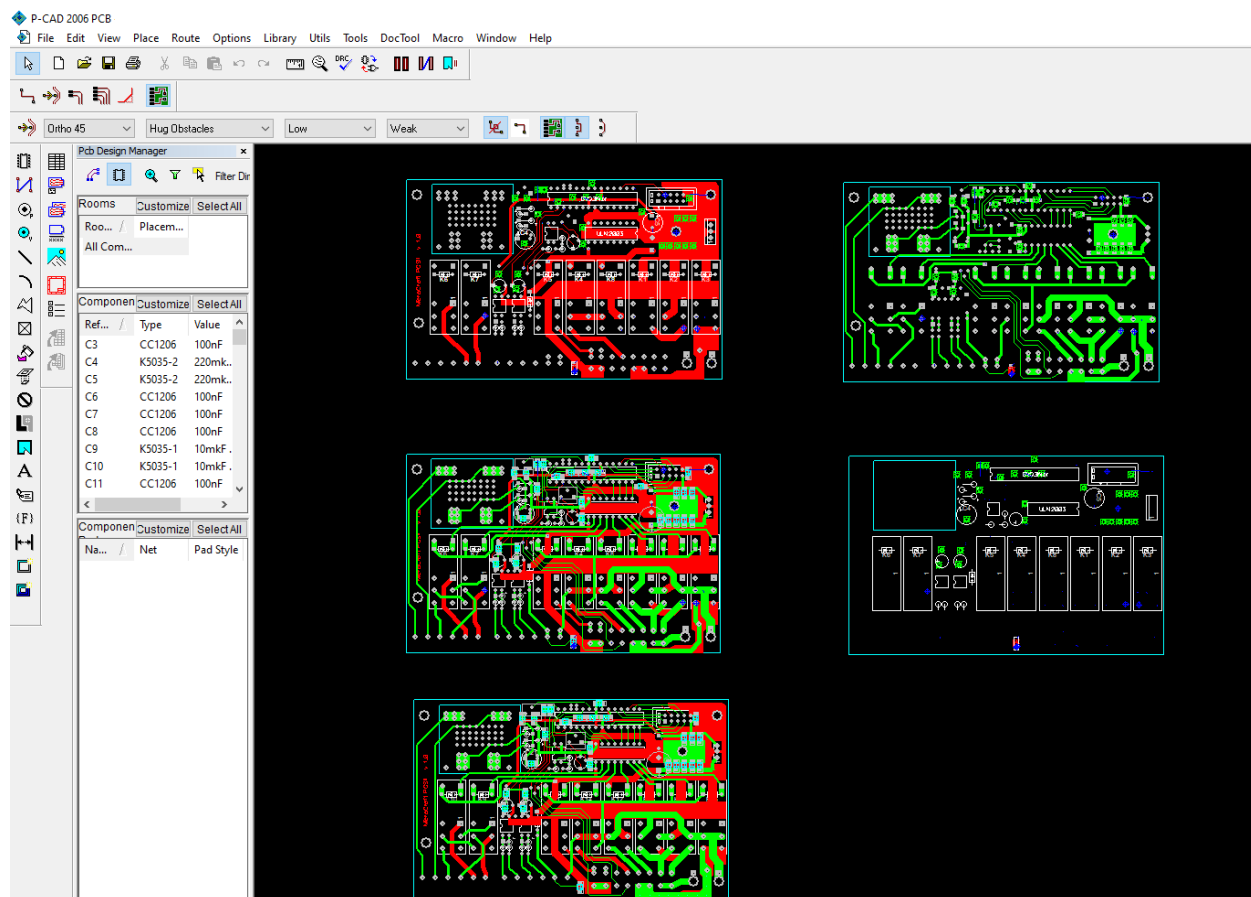
Зм.	Арк.	№ докум.	Підпис	Дата			
Розробив	Баранник Д.Є.				Контролер керування автоматичним запуском побутових електростанцій Схема підключення контролера до бензогенератора.		
Перевір.	Таран В. І.						
Н. контр.	Сімоненко В.П.						
Затверд.							
					Лім.	Арк.	Аркушів
						1	1
					КПІ ім. Ігоря Сікорського Каф. ФІОТ, гр. ІО-361		

Додаток 1

Контролер керування автоматичним запуском побутових електростанцій

Схема плати контролера у програмі PCAD 2006.

Аркушів 1



					ІАЛЦ.467100.007 Д1		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розробив		Баранник Д.Є.			Контролер керування автоматичним запуском побутових електростанцій Схема плати контролера у програмі PCAD 2006.	Лім.	Арк.
Перевір.		Таран В. І.					1
						КПІ ім. Ігоря Сікорського Каф. ФІОТ, гр. ІО-361	
Н. контр.		Сімоненко В.П.					
Затверд.							

Додаток 2

Контролер керування автоматичним запуском побутових електростанцій

Код програми РКІ екрану.

Аркушів 19

2020

/*

Библиотека для светодиодного

OLED LCD SSD1306 дисплея.

Библиотека сделана под CVAVR.

*/

#include <delay.h>

void LCD_init(void); //начальная инициализация дисплея

void LCD_Commmand(unsigned char ControByte, unsigned char DataByte);

//команды дисплею или данные

void LCD_Goto(unsigned char x, unsigned char y); //установить координаты

void LCD_Clear(void); //очистка всего дисплея

//void LCD_Contrast(char set_contrast); //настройка контраста от 0 до 255

void LCD_Blinc(unsigned int t,unsigned char x); //t секунд , x раз

void LCD_CharVeryBig(unsigned int c,unsigned char h); //

void LCD_Printf(unsigned char* buf, unsigned char size); //печатает строку с

размерами - 0 самый мелкий 2 - увеличенный - 3 самый большой (только для цифр)

void LCD_Mode(char set_mode); //1 - inverted / 0 - normal

void LCD_Sleep(char set); //1 - on sleep / 0 - off sleep

void ShowLine(unsigned char line);

void ShowBigLine(unsigned char line);

unsigned char LCD_X,LCD_Y;

#define SSD1306_I2C_ADDRESS

0x78

// size

					ІАЛЦ.467100.008 Д2			
Зм.	Арк.	№ докум.	Підпис	Дата	Контролер керування автоматичним запуском побутових електростанцій Код програми РКІ екрану.	Лім.	Арк.	Аркушів
Розробив		Баранник Д.Є.					1	19
Перевір.		Таран В. І.						
Н. контр.		Сімоненко В.П.				КПІ ім. Ігоря Сікорського Каф. ФІОТ, гр. ІО-361		
Затверд.								

```

#define SSD1306_LCDWIDTH          125
#define SSD1306_LCDHEIGHT         64
#define SSD1306_DEFAULT_SPACE     5
// Commands
#define SSD1306_SETCONTRAST        0x81
#define SSD1306_DISPLAYALLON_RESUME 0xA4
#define SSD1306_DISPLAYALLON      0xA5
#define SSD1306_NORMALDISPLAY      0xA6
#define SSD1306_INVERTDISPLAY      0xA7
#define SSD1306_DISPLAYOFF         0xAE
#define SSD1306_DISPLAYON          0xAF
#define SSD1306_SETDISPLAYOFFSET   0xD3
#define SSD1306_SETCOMPINS         0xDA
#define SSD1306_SETVCOMDETECT      0xDB
#define SSD1306_SETDISPLAYCLOCKDIV 0xD5
#define SSD1306_SETPRECHARGE       0xD9
#define SSD1306_SETMULTIPLEX       0xA8
#define SSD1306_SETLOWCOLUMN       0x00
#define SSD1306_SETHIGHCOLUMN      0x10
#define SSD1306_SETSTARTLINE       0x40
#define SSD1306_MEMORYMODE         0x20
#define SSD1306_COLUMNADDR         0x21
#define SSD1306_PAGEADDR           0x22
#define SSD1306_COMSCANINC         0xC0
#define SSD1306_COMSCANDEC         0xC8
#define SSD1306_SEGREMAP           0xA0

```

```

#define SSD1306_CHARGE_PUMP          0x8D
#define SSD1306_EXTERNAL_VCC         0x1
#define SSD1306_SWITCHCAP_VCC        0x2

// Scrolling #defines
#define SSD1306_ACTIVATE_SCROLL       0x2F
#define SSD1306_DEACTIVATE_SCROLL     0x2E
#define SSD1306_SET_VERTICAL_SCROLL_AREA 0xA3
#define SSD1306_RIGHT_HORIZONTAL_SCROLL 0x26
#define SSD1306_LEFT_HORIZONTAL_SCROLL 0x27
#define SSD1306_VERTICAL_AND_RIGHT_HORIZONTAL_SCROLL 0x29
#define SSD1306_VERTICAL_AND_LEFT_HORIZONTAL_SCROLL 0x2A

#define COMAND                        0x00
#define DATA                        0x40
flash unsigned char LCD_Buffer[0x0460] = //0x0500
{
0x00, 0x00, 0x00, 0x00, 0x00, // 00 space
0x00, 0x00, 0x5F, 0x00, 0x00, // 01 !
0x00, 0x07, 0x00, 0x07, 0x00, // 02 "
0x14, 0x7F, 0x14, 0x7F, 0x14, // 03
0x24, 0x2A, 0x7F, 0x2A, 0x12, // 04
0x23, 0x13, 0x08, 0x64, 0x62, // 05
0x36, 0x49, 0x55, 0x22, 0x50, // 06
0x00, 0x05, 0x03, 0x00, 0x00, // 07
0x00, 0x1C, 0x22, 0x41, 0x00, // 08
0x00, 0x41, 0x22, 0x1C, 0x00, // 09
0x14, 0x08, 0x3E, 0x08, 0x14, // 0A

```

					ІАЛІЦ.467100.008 Д2	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

0x08, 0x08, 0x3E, 0x08, 0x08, // 0B
 0x00, 0x50, 0x30, 0x00, 0x00, // 0C
 0x08, 0x08, 0x08, 0x08, 0x08, // 0D
 0x00, 0x60, 0x60, 0x00, 0x00, // 0E
 0x20, 0x10, 0x08, 0x04, 0x02, // 0F
 0x00, 0x00, 0x00, 0x00, 0x00, // 10
 0x00, 0x00, 0x5F, 0x00, 0x00, // 11
 0x00, 0x07, 0x00, 0x07, 0x00, // 12
 0x14, 0x7F, 0x14, 0x7F, 0x14, // 13
 0x24, 0x2A, 0x7F, 0x2A, 0x12, // 14
 0x23, 0x13, 0x08, 0x64, 0x62, // 15
 0x36, 0x49, 0x55, 0x22, 0x50, // 16
 0x00, 0x05, 0x03, 0x00, 0x00, // 17
 0x00, 0x1C, 0x22, 0x41, 0x00, // 18
 0x00, 0x41, 0x22, 0x1C, 0x00, // 19
 0x14, 0x08, 0x3E, 0x08, 0x14, // 1A
 0x08, 0x08, 0x3E, 0x08, 0x08, // 1B
 0x00, 0x50, 0x30, 0x00, 0x00, // 1C
 0x08, 0x08, 0x08, 0x08, 0x08, // 1D
 0x00, 0x60, 0x60, 0x00, 0x00, // 1E
 0x20, 0x10, 0x08, 0x04, 0x02, // 1F
 0x00, 0x00, 0x00, 0x00, 0x00, // 20 space
 0x00, 0x00, 0x5F, 0x00, 0x00, // 21 !
 0x00, 0x07, 0x00, 0x07, 0x00, // 22 "
 0x14, 0x7F, 0x14, 0x7F, 0x14, // 23 #
 0x24, 0x2A, 0x7F, 0x2A, 0x12, // 24 \$

					ІАЛЦ.467100.008 Д2	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

0x23, 0x13, 0x08, 0x64, 0x62, // 25 %
 0x36, 0x49, 0x55, 0x22, 0x50, // 26 &
 0x00, 0x05, 0x03, 0x00, 0x00, // 27 '
 0x00, 0x1C, 0x22, 0x41, 0x00, // 28 (
 0x00, 0x41, 0x22, 0x1C, 0x00, // 29)
 0x14, 0x08, 0x3E, 0x08, 0x14, // 2a *
 0x08, 0x08, 0x3E, 0x08, 0x08, // 2b +
 0x00, 0x50, 0x30, 0x00, 0x00, // 2c ,
 0x08, 0x08, 0x08, 0x08, 0x08, // 2d -
 0x00, 0x60, 0x60, 0x00, 0x00, // 2e .
 0x20, 0x10, 0x08, 0x04, 0x02, // 2f /
 0x3E, 0x51, 0x49, 0x45, 0x3E, // 30 0
 0x00, 0x42, 0x7F, 0x40, 0x00, // 31 1
 0x42, 0x61, 0x51, 0x49, 0x46, // 32 2
 0x21, 0x41, 0x45, 0x4B, 0x31, // 33 3
 0x18, 0x14, 0x12, 0x7F, 0x10, // 34 4
 0x27, 0x45, 0x45, 0x45, 0x39, // 35 5
 0x3C, 0x4A, 0x49, 0x49, 0x30, // 36 6
 0x01, 0x71, 0x09, 0x05, 0x03, // 37 7
 0x36, 0x49, 0x49, 0x49, 0x36, // 38 8
 0x06, 0x49, 0x49, 0x29, 0x1E, // 39 9
 0x00, 0x36, 0x36, 0x00, 0x00, // 3a :
 0x00, 0x56, 0x36, 0x00, 0x00, // 3b ;
 0x08, 0x14, 0x22, 0x41, 0x00, // 3c <
 0x14, 0x14, 0x14, 0x14, 0x14, // 3d =
 0x00, 0x41, 0x22, 0x14, 0x08, // 3e >
 0x02, 0x01, 0x51, 0x09, 0x06, // 3f ?

					ІАЛЦ.467100.008 Д2	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

0x32, 0x49, 0x79, 0x41, 0x3E, // 40 @
 0x7E, 0x11, 0x11, 0x11, 0x7E, // 41 A
 0x7F, 0x49, 0x49, 0x49, 0x36, // 42 B
 0x3E, 0x41, 0x41, 0x41, 0x22, // 43 C
 0x7F, 0x41, 0x41, 0x22, 0x1C, // 44 D
 0x7F, 0x49, 0x49, 0x49, 0x41, // 45 E
 0x7F, 0x09, 0x09, 0x09, 0x01, // 46 F
 0x3E, 0x41, 0x49, 0x49, 0x7A, // 47 G
 0x7F, 0x08, 0x08, 0x08, 0x7F, // 48 H
 0x00, 0x41, 0x7F, 0x41, 0x00, // 49 I
 0x20, 0x40, 0x41, 0x3F, 0x01, // 4a J
 0x7F, 0x08, 0x14, 0x22, 0x41, // 4b K
 0x7F, 0x40, 0x40, 0x40, 0x40, // 4c L
 0x7F, 0x02, 0x0C, 0x02, 0x7F, // 4d M
 0x7F, 0x04, 0x08, 0x10, 0x7F, // 4e N
 0x3E, 0x41, 0x41, 0x41, 0x3E, // 4f O
 0x7F, 0x09, 0x09, 0x09, 0x06, // 50 P
 0x3E, 0x41, 0x51, 0x21, 0x5E, // 51 Q
 0x7F, 0x09, 0x19, 0x29, 0x46, // 52 R
 0x46, 0x49, 0x49, 0x49, 0x31, // 53 S
 0x01, 0x01, 0x7F, 0x01, 0x01, // 54 T
 0x3F, 0x40, 0x40, 0x40, 0x3F, // 55 U
 0x1F, 0x20, 0x40, 0x20, 0x1F, // 56 V
 0x3F, 0x40, 0x38, 0x40, 0x3F, // 57 W
 0x63, 0x14, 0x08, 0x14, 0x63, // 58 X
 0x07, 0x08, 0x70, 0x08, 0x07, // 59 Y
 0x61, 0x51, 0x49, 0x45, 0x43, // 5a Z

					ІАЛЦ.467100.008 Д2	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

0x00, 0x7F, 0x41, 0x41, 0x00, // 5b [
 0x02, 0x04, 0x08, 0x10, 0x20, // 5c Yen Currency Sign
 0x00, 0x41, 0x41, 0x7F, 0x00, // 5d]
 0x04, 0x02, 0x01, 0x02, 0x04, // 5e ^
 0x40, 0x40, 0x40, 0x40, 0x40, // 5f _
 0x00, 0x01, 0x02, 0x04, 0x00, // 60 `
 0x20, 0x54, 0x54, 0x54, 0x78, // 61 a
 0x7F, 0x48, 0x44, 0x44, 0x38, // 62 b
 0x38, 0x44, 0x44, 0x44, 0x20, // 63 c
 0x38, 0x44, 0x44, 0x48, 0x7F, // 64 d
 0x38, 0x54, 0x54, 0x54, 0x18, // 65 e
 0x08, 0x7E, 0x09, 0x01, 0x02, // 66 f
 0x0C, 0x52, 0x52, 0x52, 0x3E, // 67 g
 0x7F, 0x08, 0x04, 0x04, 0x78, // 68 h
 0x00, 0x44, 0x7D, 0x40, 0x00, // 69 i
 0x20, 0x40, 0x44, 0x3D, 0x00, // 6a j
 0x7F, 0x10, 0x28, 0x44, 0x00, // 6b k
 0x00, 0x41, 0x7F, 0x40, 0x00, // 6c l
 0x7C, 0x04, 0x18, 0x04, 0x78, // 6d m
 0x7C, 0x08, 0x04, 0x04, 0x78, // 6e n
 0x38, 0x44, 0x44, 0x44, 0x38, // 6f o
 0x7C, 0x14, 0x14, 0x14, 0x08, // 70 p
 0x08, 0x14, 0x14, 0x18, 0x7C, // 71 q
 0x7C, 0x08, 0x04, 0x04, 0x08, // 72 r
 0x08, 0x54, 0x54, 0x54, 0x20, // 73 s
 0x04, 0x3F, 0x44, 0x40, 0x20, // 74 t
 0x3C, 0x40, 0x40, 0x20, 0x7C, // 75 u

					ІАЛЦ.467100.008 Д2	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

0x1C, 0x20, 0x40, 0x20, 0x1C, // 76 v
 0x3C, 0x40, 0x30, 0x40, 0x3C, // 77 w
 0x44, 0x28, 0x10, 0x28, 0x44, // 78 x
 0x0C, 0x50, 0x50, 0x50, 0x3C, // 79 y
 0x44, 0x64, 0x54, 0x4C, 0x44, // 7a z
 0x00, 0x08, 0x36, 0x41, 0x00, // 7b <
 0x00, 0x00, 0x7F, 0x00, 0x00, // 7c |
 0x00, 0x41, 0x36, 0x08, 0x00, // 7d >
 0x10, 0x08, 0x08, 0x10, 0x08, // 7e Right Arrow ->
 0x78, 0x46, 0x41, 0x46, 0x78, // 7f Left Arrow <-
 0x00, 0x00, 0x00, 0x00, 0x00, // 80
 0x00, 0x00, 0x5F, 0x00, 0x00, // 81
 0x00, 0x07, 0x00, 0x07, 0x00, // 82
 0x14, 0x7F, 0x14, 0x7F, 0x14, // 83
 0x24, 0x2A, 0x7F, 0x2A, 0x12, // 84
 0x23, 0x13, 0x08, 0x64, 0x62, // 85
 0x36, 0x49, 0x55, 0x22, 0x50, // 86
 0x00, 0x05, 0x03, 0x00, 0x00, // 87
 0x00, 0x1C, 0x22, 0x41, 0x00, // 88
 0x00, 0x41, 0x22, 0x1C, 0x00, // 89
 0x14, 0x08, 0x3E, 0x08, 0x14, // 8A
 0x08, 0x08, 0x3E, 0x08, 0x08, // 8B
 0x00, 0x50, 0x30, 0x00, 0x00, // 8C
 0x08, 0x08, 0x08, 0x08, 0x08, // 8D
 0x00, 0x60, 0x60, 0x00, 0x00, // 8E
 0x20, 0x10, 0x08, 0x04, 0x02, // 8F
 0x00, 0x00, 0x00, 0x00, 0x00, // 90

					ІАЛЦ.467100.008 Д2	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

0x00, 0x00, 0x5F, 0x00, 0x00, // 91
 0x00, 0x07, 0x00, 0x07, 0x00, // 92
 0x14, 0x7F, 0x14, 0x7F, 0x14, // 93
 0x24, 0x2A, 0x7F, 0x2A, 0x12, // 94
 0x23, 0x13, 0x08, 0x64, 0x62, // 95
 0x36, 0x49, 0x55, 0x22, 0x50, // 96
 0x00, 0x05, 0x03, 0x00, 0x00, // 97
 0x00, 0x1C, 0x22, 0x41, 0x00, // 98
 0x00, 0x41, 0x22, 0x1C, 0x00, // 99
 0x14, 0x08, 0x3E, 0x08, 0x14, // 9A
 0x08, 0x08, 0x3E, 0x08, 0x08, // 9B
 0x00, 0x50, 0x30, 0x00, 0x00, // 9C
 0x08, 0x08, 0x08, 0x08, 0x08, // 9D
 0x00, 0x60, 0x60, 0x00, 0x00, // 9E
 0x20, 0x10, 0x08, 0x04, 0x02, // 9F
 0x00, 0x00, 0x00, 0x00, 0x00, // A0
 0x00, 0x00, 0x5F, 0x00, 0x00, // A1
 0x00, 0x07, 0x00, 0x07, 0x00, // A2
 0x14, 0x7F, 0x14, 0x7F, 0x14, // A3
 0x24, 0x2A, 0x7F, 0x2A, 0x12, // A4
 0x23, 0x13, 0x08, 0x64, 0x62, // A5
 0x36, 0x49, 0x55, 0x22, 0x50, // A6
 0x00, 0x05, 0x03, 0x00, 0x00, // A7
 0x00, 0x1C, 0x22, 0x41, 0x00, // A8
 0x00, 0x41, 0x22, 0x1C, 0x00, // A9
 0x14, 0x08, 0x3E, 0x08, 0x14, // AA
 0x08, 0x08, 0x3E, 0x08, 0x08, // AB

					ІАЛЦ.467100.008 Д2	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

0x00, 0x50, 0x30, 0x00, 0x00, // AC
 0x08, 0x08, 0x08, 0x08, 0x08, // AD
 0x00, 0x60, 0x60, 0x00, 0x00, // AE
 0x20, 0x10, 0x08, 0x04, 0x02, // AF
 // 0x3E, 0x51, 0x49, 0x45, 0x3E, // B0
 0x00, 0x06, 0x09, 0x09, 0x06,
 0x00, 0x42, 0x7F, 0x40, 0x00, // B1
 0x42, 0x61, 0x51, 0x49, 0x46, // B2
 0x21, 0x41, 0x45, 0x4B, 0x31, // B3
 0x18, 0x14, 0x12, 0x7F, 0x10, // B4
 0x27, 0x45, 0x45, 0x45, 0x39, // B5
 0x3C, 0x4A, 0x49, 0x49, 0x30, // B6
 0x01, 0x71, 0x09, 0x05, 0x03, // B7
 0x36, 0x49, 0x49, 0x49, 0x36, // B8
 0x06, 0x49, 0x49, 0x29, 0x1E, // B9
 0x00, 0x36, 0x36, 0x00, 0x00, // BA
 0x00, 0x56, 0x36, 0x00, 0x00, // BB
 0x08, 0x14, 0x22, 0x41, 0x00, // BC
 0x14, 0x14, 0x14, 0x14, 0x14, // BD
 0x00, 0x41, 0x22, 0x14, 0x08, // BE
 0x02, 0x01, 0x51, 0x09, 0x06, // BF
 0x7E, 0x11, 0x11, 0x11, 0x7E, // C0 A
 0x7F, 0x49, 0x49, 0x49, 0x31, // C1 Б
 0x7F, 0x49, 0x49, 0x49, 0x36, // C2 B
 0x7F, 0x01, 0x01, 0x01, 0x03, // C3 Г
 0x60, 0x3E, 0x21, 0x21, 0x7F, // C4 Д
 0x7F, 0x49, 0x49, 0x49, 0x41, // C5 E

					ІАЛЦ.467100.008 Д2	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

0x77, 0x08, 0x7F, 0x08, 0x77, // C6 Ж
 0x22, 0x41, 0x49, 0x49, 0x36, // C7 З
 0x7F, 0x10, 0x08, 0x04, 0x7F, // C8 И
 0x7F, 0x10, 0x09, 0x04, 0x7F, // C9 И
 0x7F, 0x08, 0x14, 0x22, 0x41, // CA К
 0x40, 0x3E, 0x01, 0x01, 0x7F, // CB Л
 0x7F, 0x02, 0x0C, 0x02, 0x7F, // CC М
 0x7F, 0x08, 0x08, 0x08, 0x7F, // CD Н
 0x3E, 0x41, 0x41, 0x41, 0x3E, // CE О
 0x7F, 0x01, 0x01, 0x01, 0x7F, // CF П
 0x7F, 0x09, 0x09, 0x09, 0x06, // D0 Р
 0x3E, 0x41, 0x41, 0x41, 0x22, // D1 С
 0x01, 0x01, 0x7F, 0x01, 0x01, // D2 Т
 0x27, 0x48, 0x48, 0x48, 0x3F, // D3 У
 0x1E, 0x21, 0x7F, 0x21, 0x1E, // D4 Ф
 0x63, 0x14, 0x08, 0x14, 0x63, // D5 X
 0x3F, 0x20, 0x20, 0x3F, 0x60, // D6 Ц
 0x07, 0x08, 0x08, 0x08, 0x7F, // D7 Ч
 0x7F, 0x40, 0x7F, 0x40, 0x7F, // D8 Ш
 0x3F, 0x20, 0x3F, 0x20, 0x7F, // D9 Щ
 0x01, 0x7F, 0x48, 0x48, 0x30, // DA Ъ
 0x7F, 0x48, 0x30, 0x00, 0x7F, // DB Ы
 0x00, 0x7F, 0x48, 0x48, 0x30, // DC Ь
 0x22, 0x41, 0x49, 0x49, 0x3E, // DD Э
 0x7F, 0x08, 0x3E, 0x41, 0x3E, // DE Ю
 0x46, 0x29, 0x19, 0x09, 0x7F // DF Я
 };

					ІАЛЦ.467100.008 Д2	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

```

void LCD_init(void)
{
    #asm("cli")
    LCD_Sleep(0);
    delay_ms(10);
    LCD_Commmand(COMAND, SSD1306_SETDISPLAYCLOCKDIV);
    LCD_Commmand(COMAND, 0x80);
    LCD_Commmand(COMAND, SSD1306_SETMULTIPLEX);
    LCD_Commmand(COMAND, 0x3F);
    LCD_Commmand(COMAND, SSD1306_SETDISPLAYOFFSET);
    LCD_Commmand(COMAND, 0x00);
    LCD_Commmand(COMAND, SSD1306_SETSTARTLINE | 0x00);
    // We use internal charge pump
    LCD_Commmand(COMAND, SSD1306_CHARGEPUMP);
    LCD_Commmand(COMAND, 0x14);
    // Horizontal memory mode
    LCD_Commmand(COMAND, SSD1306_MEMORYMODE);
    LCD_Commmand(COMAND, 0x00);
    LCD_Commmand(COMAND, SSD1306_SEGREMAP | 0x1);
    LCD_Commmand(COMAND, SSD1306_COMSCANDEC);
    LCD_Commmand(COMAND, SSD1306_SETCOMPINS);
    LCD_Commmand(COMAND, 0x12);
    // Max contrast
    LCD_Commmand(COMAND, SSD1306_SETCONTRAST);
    LCD_Commmand(COMAND, 0xCF);
    LCD_Commmand(COMAND, SSD1306_SETPRECHARGE);

```

```

LCD_Commmand(COMAND, 0xF1);
LCD_Commmand(COMAND, SSD1306_SETVCOMDETECT);
LCD_Commmand(COMAND, 0x40);
LCD_Commmand(COMAND, SSD1306_DISPLAYALLON_RESUME);
    // Non-inverted display
LCD_Mode(0);
    // Turn display back on
LCD_Sleep(1);

LCD_Clear();
LCD_Goto(0,0);
#asm("sei")
}

void LCD_Mode(char set_mode)
{
    if(set_mode==0){ LCD_Commmand(COMAND,
SSD1306_NORMALDISPLAY); }
    if(set_mode==1){ LCD_Commmand(COMAND, SSD1306_INVERTDISPLAY);
}
}

void LCD_Sleep(char set)
{
    if(set==0){LCD_Commmand(COMAND,SSD1306_DISPLAYOFF);}
    if(set==1){LCD_Commmand(COMAND,SSD1306_DISPLAYON);}
}

```

```

void LCD_Commmand(unsigned char ControByte, unsigned char DataByte)
{
    #asm("cli")
    i2c_start();
    i2c_write(SSD1306_I2C_ADDRESS);
    i2c_write(ControByte);
    i2c_write(DataByte);
    i2c_stop();
    #asm("sei")
}

```

```

void LCD_Goto(unsigned char x, unsigned char y)
{
    #asm("cli")
        LCD_X = x;
        LCD_Y = y;

    // установка и настройка
    LCD_Commmand(COMAND, 0xB0 + y); //установить адрес начала
координат
    LCD_Commmand(COMAND, x & 0xf); //установить нижний адрес
столбца
    LCD_Commmand(COMAND, 0x10 | (x >> 4)); //установить верхний адрес
столбца
    #asm("sei")
}

```

					ІАЛЦ.467100.008 Д2	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

```

void LCD_Clear(void)
{
    unsigned short i;
    unsigned short x=0;
    unsigned short y=0;
    #asm("cli")
    LCD_Goto(0,0);

    for (i=0; i<((SSD1306_LCDWIDTH*SSD1306_LCDHEIGHT/8); i++)
    //(SSD1306_LCDWIDTH*SSD1306_LCDHEIGHT/8)
    {
        LCD_CharVeryBig(' ',0);
        x ++;
        if(x>SSD1306_LCDWIDTH)
        {
            x =0;
            y++;
            LCD_Goto(0,y);
        }
    }
    LCD_X =SSD1306_DEFAULT_SPACE;
    LCD_Y =0;
    #asm("sei");
}

void LCD_Blinc(unsigned int t,unsigned char x )
{

```

```

unsigned char xx = 0;
for (xx=0; xx<x; xx++)
{
    LCD_Mode(1);
    delay_ms(t);
    LCD_Mode(0);
    delay_ms(t);
}
}

void LCD_CharVeryBig(unsigned int c,unsigned char h)
{
    unsigned char x = 0;
    unsigned int m = 0;
    unsigned char cl = 0;
    #asm("cli")
    i2c_start();
    i2c_write(SSD1306_I2C_ADDRESS);
    i2c_write(DATA);//data mode
    for (x=0; x<5; x++)
    {
        //0x27, 0x48, 0x48, 0x48, 0x3F,// D3 Y

        m = LCD_Buffer[c*5+x];
        cl=m;
        if(h==1)
        {

```

					ІАЛЦ.467100.008 Д2	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16


```

i2c_write(cl);
}
else
if(h==2)
{
    cl = 0;
    if (0b00000001 == (m & 0b00000001)) { cl =cl+0b000000011;};
    if (0b00000010 == (m & 0b00000010)) { cl =cl+0b00001100;};
    if (0b00000100 == (m & 0b00000100)) { cl =cl+0b00110000;};
    if (0b00001000 == (m & 0b00001000)) { cl =cl+0b11000000;};
    i2c_write(cl);
}
else if(h==3)
{
    cl = 0;
    if (0b00010000 == (m & 0b00010000)) { cl =cl+0b000000011;};
    if (0b00100000 == (m & 0b00100000)) { cl =cl+0b00001100;};
    if (0b01000000 == (m & 0b01000000)) { cl =cl+0b00110000;};
    if (0b10000000 == (m & 0b10000000)) { cl =cl+0b11000000;};
    i2c_write(cl);
}
i2c_write(cl);
}

```

```

i2c_write(0x00); //пробел в 1 точку между символами
// i2c_write(0x00); //пробел в 1 точку между символами
i2c_stop(); // stop transmitting

```

```

        if(h==0){
            LCD_X += 6;
        }
        else{
            LCD_X += 12;
        }
        #asm("sei")
    }

void LCD_Printf(unsigned char* buf, unsigned char size) //выводим строку из
буфера
{

    while ((*buf!=0)&&(LCD_X<SSD1306_LCDWIDTH))
    {
        //
        if((LCD_X>SSD1306_LCDWIDTH)||(LCD_X<5)){LCD_X=SSD1306_DEFAULT_SPACE;}
        LCD_CharVeryBig(*buf++,size);
    }

}

//+++++
+++++

void ShowLine(unsigned char line)
{

```

```

LCD_Goto(0,line);

LCD_Printf(buff,0); // вывод на дисплей
}

//+++++
+++++

void ShowBigLine(unsigned char line)
{
    LCD_Goto(0,line);
    LCD_Printf(buff,2); // вывод на дисплей
    LCD_Goto(0,line+1);
    LCD_Printf(buff,3); // вывод на дисплей
}

```

					ІАЛЦ.467100.008 Д2	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

Додаток 3

Контролер керування автоматичним запуском побутових електростанцій

Код програми контролера.

Аркушів 30

2020

/******

This program was created by the
CodeWizardAVR V3.12 Advanced
Automatic Program Generator

Project : KPI Bachelor

Version : 1

Date : 09.03.2020

Author : Barannyk Danyl

Company :

Comments:

Chip type : ATmega8

Program type : Application

AVR Core Clock frequency: 1,000000 MHz (8 MHz)

Memory model : Small

External RAM size : 0

Data Stack size : 256

*****/

static char buff[30];

#include <mega8.h>

#include <delay.h>

#include <i2c.h>

#include <stdio.h>

#include "ssd1306.c"

// Voltage Reference: Int., cap. on AREF

					ІАЛЦ.467100.009 ДЗ			
Зм.	Арк.	№ докум.	Підпис	Дата	Контролер керування автоматичним запуском побутових електростанцій Код програми контролера.	Лім.	Аркуш	Аркушів
Розробив		Баранник Д.Є.					1	30
Перевір.		Таран В. І.						
Н. контр.		Сімоненко В.П.				КПІ ім. Ігоря Сікорського Каф. ФІОТ, гр. ІО-361		
Затверд.								

```

#define ADC_VREF_TYPE ((1<<REFS1) | (1<<REFS0) | (0<<ADLAR))

//#define PWR_LCD      PORTD.4
#define STARTER_REL    PORTB.0

#define GEN_REL        PORTD.2
#define LINE_REL       PORTD.3
#define ON_REL         PORTD.4
#define OFF_REL        PORTD.5
#define VALVE_U_REL    PORTD.6
#define VALVE_D_REL    PORTD.7

#define ROTOR_PIN      PINC.4
#define GEN_OUT_PIN    PIND.0

#define KEY_E          PINB.4
#define KEY_U          PINB.3
#define KEY_D          PINB.2
#define KEY_SIG_NUM    10
#define GO_TO_OPTIONS  20
#define INFOLINE1      6
#define INFOLINE2      7
#define LAST_MODE_MENU  3
#define FIRST_OPTION_MENU 10  //
#define LAST_OPTION_MENU 16  //

#define U1_MIN          140  //
#define U1_MAX          265  //

```

					ІАЛІЦ.467100.009 ДЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		2

```
#define ONE_SECOND      4000  //
#define HALF_SECOND     2000  //
#define QUARTER_SECOND  1000  //
#define NEXT_ZAPUSK_SECOND 20000 // 5с
```

```
#define RAZGON          5      // 5с
#define PODSVETKA       60
```

```
// Global variables
```

```
eeprom unsigned char Termo[256]=      // +50C
```

```
{175,170,155,145,135,130,125,120,117,115,
112,110,107,105,104,102,100, 99, 98, 96,
95, 94, 93, 91, 90, 89, 88, 87, 86, 85,
84, 84, 83, 82, 82, 81, 80, 80, 79, 78,
78, 77, 77, 76, 76, 75, 75, 74, 74, 73,
72, 72, 71, 71, 70, 70, 69, 69, 68, 68,
68, 67, 67, 67, 66, 66, 65, 65, 65, 64,
64, 64, 63, 63, 63, 62, 62, 62, 61, 61,
61, 60, 60, 60, 60, 59, 59, 59, 58, 58,
58, 58, 57, 57, 57, 56, 56, 56, 55, 55,
55, 55, 55, 54, 54, 54, 54, 53, 53, 53,
53, 53, 52, 52, 52, 52, 51, 51, 51, 51,
50, 50, 50, 50, 49, 49, 49, 49, 49, 49,
48, 48, 48, 48, 48, 47, 47, 47, 47, 47,
47, 46, 46, 46, 46, 46, 45, 45, 45, 45,
45, 44, 44, 44, 44, 44, 44, 43, 43, 43,
43, 43, 43, 43, 42, 42, 42, 42, 42, 41,
41, 41, 41, 41, 41, 40, 40, 40, 40 ,40,
40, 39, 39, 39, 39, 39, 39, 39, 38, 38,
38, 38, 38, 38, 38, 37, 37, 37, 37, 37,
```

					ІАЛІЦ.467100.009 ДЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

37, 37, 36, 36, 36, 36, 35, 35, 35,
 35, 35, 35, 35, 34, 34, 34, 34, 34, 34,
 34, 33, 33, 33, 33, 33, 33, 33, 33, 32,
 32, 32, 32, 32, 32, 32, 31, 31, 31, 31,
 31, 31, 31, 30, 30, 30, 30, 30, 30, 30,
 30, 29, 29, 29, 20, 20};

//

В МЕНЮ!

EEPROM unsigned char typeEEP = 0; // тип

електростанції

EEPROM unsigned char t1EEP = 10; // час переходу на

генератор EEPROM unsigned char t2EEP = 10; // час

переходу на мережу

EEPROM unsigned char t3EEP = 6; // час роботи

стартера

EEPROM unsigned char t4EEP = 8; // час роботи із

заслонкою

EEPROM unsigned char t5EEP = 5; // час без навантаження (прогрів в
 нормальном режимі)

// Охолодження - в 2 рази більше ніж прогрів без

навантаження

//EEPROM unsigned char t6EEP = 10; // час охолодження

EEPROM unsigned char TminZEEP = 20; // Мінімальна температура

прогріву генератора с заслонкой

EEPROM unsigned char TminNEEP = 50; // Мінімальна температура

прогріву генератора -

EEPROM unsigned char Z_EEP = 0; //

zaslonka/klapan

EEPROM unsigned char R_EEP = 0; // Перевірка сигналу "Обертання
 ротора"

					ІАЛЦ.467100.009 ДЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

EEPROM unsigned char KilistZapuskiv_EEP=3; // Кількість спроб
запуску

EEPROM unsigned int NarabotkaEEP=0;
EEPROM unsigned char TO_EEP=1; //ТО заноситься 1 якщо пора робити
ТО, можна скинути із меню

static unsigned char
type,t1,t2,t3,t4,t5,TminZ,TminN,KilistZapuskiv;//,Narabotka,;
static unsigned char AvtRu=0; //0-Auto; 1-Ru4noy
static bit z=0;
static bit r=0;
//static bit rbit=0;
static bit TO_output=0;

static unsigned char up,down,enter;
static unsigned char PositionPointer = 1;
static bit podmenu=0;

static unsigned char podsvet=60; //sec
static unsigned int U_Bat=0;
static unsigned int U1;
static int T;

static unsigned char rotor= 0;
//static unsigned char average_ADC_counter=0;
static unsigned int interrapt_counter=0;
static unsigned char second_counter=0;
static unsigned int minut_counter=0;

					ІАЛЦ.467100.009 ДЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

```

unsigned int U_Bat_ADC=0;
unsigned int T_ADC=0;
unsigned int U1_ADC=0;

static unsigned int temp;

//char buff[30]; //буфер дисплея

//+++++
+++++
// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | ADC_VREF_TYPE;
    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=(1<<ADSC);
    // Wait for the AD conversion to complete
    while ((ADCSRA & (1<<ADIF))==0);
    ADCSRA|=(1<<ADIF);
    return ADCW;
}

//+++++
+++++

//+++++
+++++

void UpDownEnterKey()    //перенести в переривання!
{

```

					ІАЛЦ.467100.009 ДЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

```

//Main menu
if((PositionPointer>=1)&(PositionPointer<=LAST_MODE_MENU))
{
    if((up==1)|(up>KEY_SIG_NUM)){PositionPointer++;up=2;};
    if((down==1)|(down>KEY_SIG_NUM)){PositionPointer--;down=2;};

    if(PositionPointer>LAST_MODE_MENU){PositionPointer=LAST_MODE_M
ENU;} //limits of Main menu
    if(PositionPointer<1){PositionPointer=1;}
    // Go to Options menu
    if((enter>GO_TO_OPTIONS)&(PositionPointer==LAST_MODE_MENU)){Po
sitionPointer=FIRST_OPTION_MENU;enter=2;LCD_Clear();}; //в меню
    if((enter>GO_TO_OPTIONS)&(PositionPointer==2)){PositionPointer=30;enter
=2;KilkistZapuskiv=KilkistZapuskiv_EEP;}; //на работу

}
if((PositionPointer>=FIRST_OPTION_MENU)&(PositionPointer<=LAST_OPTI
ON_MENU))
{
    if(podmenu == 0)
    {
        if((up==1)|(up>KEY_SIG_NUM)){PositionPointer++;up=2;};
        if((down==1)|(down>KEY_SIG_NUM)){PositionPointer--;down=2;};
    }

    if((enter==1)&(PositionPointer!=LAST_OPTION_MENU))
    {

        podmenu=!podmenu;
        enter=2;
    }
}

```

```

        if(podmenu==0){
            #asm("cli")
            typeEEP=type;
            t1EEP=t1;
            t2EEP=t2;
            t3EEP=t3;
            t4EEP=t4;
            t5EEP=t5;
            TminZEEP=TminZ;
            TminNEEP=TminN;
            KilkistZapuskiv_EEP=KilkistZapuskiv;
            Z_EEP=z;
            R_EEP=r;
            #asm("sei");
        }
    }

    if(PositionPointer>LAST_OPTION_MENU){PositionPointer=FIRST_OPTION_MENU;} //limits of Options menu

    if(PositionPointer<FIRST_OPTION_MENU){PositionPointer=LAST_OPTION_MENU;}

    if((enter>GO_TO_OPTIONS)&(PositionPointer==LAST_OPTION_MENU))
    {PositionPointer=LAST_MODE_MENU;enter=2;LCD_Clear(); }
    }
}

//+++++
+++++

// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{

```

					ІАЛЦ.467100.009 ДЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

#asm("cli")

if(rotor>8){STARTER_REL=0;}//ротор працює- стартер не потрібен
відключаємо стартер v1.4)

if(ROTOR_PIN==0){rotor=rotor+1;}else{rotor=rotor-1;};

if(rotor<1) rotor=1;

if(rotor>10) rotor=10;

if(KEY_U==0){up=up+1;podsvet=PODSVETKA;} else {up=0;};

if(KEY_D==0){down=down+1;podsvet=PODSVETKA;} else {down=0;};

if(KEY_E==0){enter=enter+1;podsvet=PODSVETKA;}else {enter=0}; //bee
p begin -DDRD.2=1;

if(up>99) up=99;

if(down>99) down=99;

if(enter>99) enter=99;

UpDownEnterKey();

/*

U_Bat_ADC=read_adc(0);

U1_ADC=read_adc(2);

T_ADC=read_adc(1);

T_ADC=T_ADC/4;

T=Termo[T_ADC];

T=T-50;

U_Bat=(U_Bat_ADC*20)/102; //останній розряд- десяті !!!!при виводі
ділити на 10

U1=(U1_ADC*11)/41;//ціле без десятих

*/

U_Bat_ADC=U_Bat_ADC+read_adc(1);

					ІАЛЦ.467100.009 ДЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

```

U_Bat_ADC=U_Bat_ADC>>1;
U1_ADC=U1_ADC+read_adc(0);
U1_ADC=U1_ADC>>1;

T_ADC=T_ADC+read_adc(5);
T_ADC=T_ADC>>1;

if(interrapt_counter>=31) //61~2sec - 8 mHz
{
    U_Bat=U_Bat_ADC/5; //останній розряд - десяті! При виводі ділити на 10
    U1=(U1_ADC*11)/41;//ціле без десятих
    T=Termo[T_ADC>>2]; // divide 4
    T=T-50;

    interrapt_counter=0;
    second_counter++;
    if(podsvet>0)podsvet--;

    ////////////ТО
    if(TO_EEP!=0)
    {
        if (second_counter%2!=0){TO_output=1; }
        else{TO_output=0;};//БЛИМАННЯ ТО
    }
    ////////////ТО
    if(second_counter>=59)
    {
        second_counter=0;
        podsvet=10;
    }
}

```

					ІАЛЦ.467100.009 ДЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

```

        if(PositionPointer==32){minut_counter++;};
    }
}

interrapt_counter++;
#asm("sei")
}

//+++++
+++++

void PerehodNaGenerator()
{
    LINE_REL=1; // відключаємо МЕРЕЖУ
    delay_ms(HALF_SECOND);
    GEN_REL=1; // Підключаємо генератор
}

//+++++
+++++

void PerehodNaMereju()
{
    GEN_REL=0; // відключаємо ГЕНЕРАТОР
    delay_ms(HALF_SECOND);
    LINE_REL=0; // Підключаємо МЕРЕЖУ
}

//+++++
+++++

void MainInfo()
{
    sprintf(buff," Uin =%4.uB ",U1);ShowBigLine(3);
    sprintf(buff,"      Ub = %2.u.%1.uB      ",U_Bat/10,U_Bat%10);ShowLine(5);
    if(TO_EEP)
    {

```

```

if(TO_output)
{
    sprintf(buff,"          ");
}
else
{
    sprintf(buff," !!! ОБСЛУЖИВАНИЕ !!! ");

}
ShowLine(INFOLINE2);
}

if (T<0){ sprintf(buff," T =-%3.i*C N:%5.uh ",-T,NarabotkaEEP/6);}
else{ sprintf(buff," T = %3.i*C N:%5.uh ",T,NarabotkaEEP/6);};
if (T<-25){sprintf(buff," T = n/c  N:%5.uh
",NarabotkaEEP/6);}; ShowLine(6);
}

//+++++
+++++
//----- MAIN -----
-----

//+++++
+++++

void main(void)
{
// Declare your local variables here
// Input/Output Ports initialization

PORTB=0b00000000; DDRB =0b00100011;    // 0-START ; 1,5 - OLED; 2,3,4
-keys; 6,7 - zaslonka

```



```

PORTC=0b00000000; DDRC =0b00000000;    //
PORTD=0b00000000; DDRD =0b11111100;

// Timer/Counter 0 initialization
// Clock source: System Clock; Clock value: 0,977 kHz ???
TCCR0=(1<<CS02) | (0<<CS01) | (1<<CS00);
TCNT0=0x00;
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) |
(0<<OCIE1B) | (0<<TOIE1) | (1<<TOIE0);
// External Interrupt(s) initialization  INT0: Off; INT1: Off
MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
// USART initialization USART disabled
UCSRB=(0<<RXCIEN) | (0<<TXCIEN) | (0<<UDRIEN) | (0<<RXEN) | (0<<TXEN) |
(0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);
// Analog Comparator: Off
ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) |
(0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);
// ADC Clock frequency: 1000.000 kHz, AREF pin
ADMUX=ADC_VREF_TYPE;
ADCSRA=(1<<ADEN) | (0<<ADSC) | (0<<ADFR) | (0<<ADIF) | (0<<ADIE) |
(0<<ADPS2) | (0<<ADPS1) | (1<<ADPS0);
SFIOR=(0<<ACME);

// Bit-Banged I2C Bus initialization
// Project|Configure|C Compiler|Libraries|I2C menu.
////////////////////////////////////
// Init Variables and sequenses

PositionPointer=1;

```

```

//Narabotka=0;

// завантажуюємо із EEPROM
type=typeEEP;
t1=t1EEP;
t2=t2EEP;
t3=t3EEP;
t4=t4EEP;
t5=t5EEP;
//t6=t6EEP;

TminZ=TminZEEP;
TminN=TminNEEP;

z=Z_EEP;
r=R_EEP;

KilkistZapuskiv=KilkistZapuskiv_EEP;

//EEPROM


U1=220;
i2c_init();

#asm("sei") // Global enable interrupts

// PWR_LCD=1;

////////////////////////////////////

LCD_init();

sprintf(buff," MicroCraft "); ShowBigLine(0);
LCD_Goto(45,3);


sprintf(buff,"(TM)");
LCD_Printf(buff,1); // вивід на дисплей
LCD_Goto(0,6);
sprintf(buff," PGSII v.1 ");

```

					ІАЛЦ.467100.009 ДЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

```

LCD_Printf(buff,1); // вивід на дисплей

LCD_Blinc(QUARTER_SECOND,3);

while (1)
{
    #asm("cli")
    if(podmenu == 0){LCD_Mode(0);}
    else{LCD_Mode(1);}
    #asm("sei")

    if (minut_counter>=10)
    {
        #asm("cli")
        NarobotkaEEP++;
        minut_counter=0;
        if
        ((NarobotkaEEP==120)|(NarobotkaEEP==600)|(NarobotkaEEP==1200)|(Narobotk
aEEP==1800)|(NarobotkaEEP==2400)){ NarobotkaEEP++;TO_EEP=1;};
        #asm("sei")
    }; // для підрахунку 10хв
    //////////////////////////////////////
    switch (PositionPointer){
    //////////////////////////////////////
        case 1: // автоматичний запуск
            //if(((up==1)|(up>KEY_SIG_NUM))&(podmenu==1)){#asm("cli")
TO_EEP=0; #asm("sei") up=2;}
            if(enter>=98){enter=2;LCD_Clear();#asm("cli") TO_EEP=0; #asm("sei")
up=2; }

```

```

    AvtRu=0; // встановлюємо флаг, який буде показувати який був
останній режим - Автомат чи Ручний
    if(podsvet>=1){
        sprintf(buff," 1: АВТО  ");
        ShowBigLine(0);
        MainInfo();
//    if(podsvet==1){LCD_Clear();podsvet=0;}
    }
    else
    {
        LCD_Clear();
        sprintf(buff,"$");
        ShowBigLine(0);
        delay_ms(100);
        LCD_init();
    }

    if ((U1<U1_MIN)||((U1>U1_MAX)){ //напруга в мережі низька 175В-
80% або висока 265
//додан швидкий захист навантаження
        if(U1>U1_MAX){LINE_REL=1;} // відключаємо МЕРЕЖУ для
захисту!
        temp=t1; // завантажуюємо затримку переходу на генератор
        while (temp!=0)
        {
            sprintf(buff,"  НЕТ СЕТИ! "); ShowBigLine(3);
            sprintf(buff,"   %3.1uc  ",temp);ShowBigLine(5);
            delay_ms(ONE_SECOND);
            if(PositionPointer!=1){goto begin;}
            temp--;

```

					ІАЛЦ.467100.009 ДЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

```

    }
    if ((U1<U1_MIN-5)||((U1>U1_MAX+5)){PositionPointer=30;
KilkistZapuskiv=KilkistZapuskiv_EEP;};

    //напруга все ще за допустимими межами ?,тоді перехід.
    // доданий гистерезис
    }

break;

////////////////////////////////////

case 2: //

    AvtRu=1; // встановлюємо флаг, який буде показувати який був
останній режим - Автомат чи Ручний
    sprintf(buff," 2: РУЧНОЙ ");          ShowBigLine(0);
    MainInfo();
    if (U1>U1_MAX)
    {
        PositionPointer=42;// перехід на аварію
        LINE_REL=1; // відключаємо МЕРЕЖУ   !!!
        LCD_Clear();
    }

break;

////////////////////////////////////

case 3: //

    sprintf(buff," 3: СТОП  ");          ShowBigLine(0);
    MainInfo();

    if (U1>U1_MAX)
    {
        PositionPointer=42;// перехід на аварію
        LINE_REL=1; // відключаємо
МЕРЕЖУ   !!!

```

					ІАЛЦ.467100.009 ДЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

```

        LCD_Clear();

    }

    break;

////////////////////////////////////

    case 10: // час роботи із заслонкою

        sprintf(buff," НАСТРОЙКИ ");      ShowBigLine(0);
        sprintf(buff,"                "); ShowLine(5);
        sprintf(buff," t4  =%4.uc ",t4); ShowBigLine(3);
        sprintf(buff," ВРЕМЯ ОБОГАЩЕНИЯ "); ShowLine(INFOLINE1);
//    sprintf(buff,"    СМЕСИ    "); ShowLine(INFOLINE2);
        if(((up==1)|(up>KEY_SIG_NUM))&(podmenu==1)){t4++;up=2;}
        if(((down==1)|(down>KEY_SIG_NUM))&(podmenu==1)){t4--;down=2;}
        if(t4>60){t4=60;}
        if(t4<1){t4=1;}

        break;

////////////////////////////////////

    case 11: //  t5=t6

        sprintf(buff," t5  =%4.uc ",t5);      ShowBigLine(3);
//    sprintf(buff," ВРЕМЯ ПРОГРЕВА  "); ShowLine(INFOLINE1);
        sprintf(buff," ПРОГРЕВ БЕЗ НАГРУЗКИ "); ShowLine(INFOLINE1);
        if(((up==1)|(up>KEY_SIG_NUM))&(podmenu==1)){t5=t5+5;up=2;}
        if(((down==1)|(down>KEY_SIG_NUM))&(podmenu==1)){t5=t5-
5;down=2;}
        if(t5>60){t5=60;}
        if(t5<5){t5=5;}

        break;

////////////////////////////////////

    case 12: // Мінімальна температура прогріву генератора с
заслонкой(накал свечей для дизеля)

        sprintf(buff," Tz  =%3.1*C ",TminZ);    ShowBigLine(3);

```

```

//      sprintf(buff,"  Max. ТЕМПЕРАТУРА  "); ShowLine(INFOLINE1);
//      sprintf(buff,"  ОБОГАЩЕНИЯ СМЕСИ  "); ShowLine(INFOLINE2);
      sprintf(buff," МИН.ТЕМП.ОБОГАЩЕНИЯ  "); ShowLine(INFOLINE1);
      if(((up==1)|(up>KEY_SIG_NUM))&(podmenu==1)){ TminZ=TminZ+5;up
=2;}

      if(((down==1)|(down>KEY_SIG_NUM))&(podmenu==1)){ TminZ=Tmin
Z-5;down=2;}

      if(TminZ>90){ TminZ=90;}
      if(TminZ<5){ TminZ=5;}

      break;

////////////////////////////////////

      case 13: //

      sprintf(buff," Kz  =%4.у ",KilkistZapuskiv); ShowBigLine(3);
//      sprintf(buff,"  КОЛИЧЕСТВО ПОПЫТОК  "); ShowLine(INFOLINE1);
//      sprintf(buff,"  ЗАПУСКА ЭЛ.СТАНЦИИ  "); ShowLine(INFOLINE2);
      sprintf(buff," КОЛИЧЕСТВО ЗАПУСКОВ  "); ShowLine(INFOLINE1);
      if(((up==1)|(up>KEY_SIG_NUM))&(podmenu==1)){ KilkistZapuskiv++;u
p=2;}

      if(((down==1)|(down>KEY_SIG_NUM))&(podmenu==1)){ KilkistZapuski
v--;down=2;}

      if(KilkistZapuskiv>6){ KilkistZapuskiv=6;}
      if(KilkistZapuskiv<1){ KilkistZapuskiv=1;}

      break;

////////////////////////////////////

      case 14: //

      if(z==0){ sprintf(buff," <ЗАСЛОНКА> ");}
      else{ sprintf(buff," < КЛАПАН > ");}

      ShowBigLine(3);

      sprintf(buff,"  ТИП ПРИВОДА  "); ShowLine(INFOLINE1);
//      sprintf(buff,"  И ИМПУЛЬСА  "); ShowLine(INFOLINE2);

```

					ІАЛЦ.467100.009 ДЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

```

        if(((up==1)|(up>KEY_SIG_NUM))&(podmenu==1)){z=!z;up=2;}
        if(((down==1)|(down>KEY_SIG_NUM))&(podmenu==1)){z=!z;down=2;}
    }

    break;

    //////////////////////////////////////

    case 15: //
        if(r==0){sprintf(buff," POTOП(off)");}
        else{sprintf(buff," POTOП(on) ");}
        ShowBigLine(3);
//        sprintf(buff,"    КОНТРОЛЬ    "); ShowLine(INFOLINE1);
        sprintf(buff," СИГНАЛ : 'POTOП' "); ShowLine(INFOLINE1);
        if(((up==1)|(up>KEY_SIG_NUM))&(podmenu==1)){r=!r;up=2;}
        if(((down==1)|(down>KEY_SIG_NUM))&(podmenu==1)){r=!r;down=2;}
        break;

    case 16: //
        sprintf(buff," - ВЫХОД - ");        ShowBigLine(3);
//        sprintf(buff,"    ДЛЯ ВЫХОДА    "); ShowLine(INFOLINE1);
//        sprintf(buff,"    УДЕРЖИВАЙ 'ВВОД' "); ShowLine(INFOLINE2);
        sprintf(buff,"    ДЛЯ ВЫХОДА >'ВВОД' "); ShowLine(INFOLINE1);

//        sprintf(buff,"%2.u",enter);
//        ShowLine(4);

        break;

    //////////////////////////////////////

    case 30: //
        LCD_Clear();
        sprintf(buff," ЗАПУСК %1.u ",KilkistZapuskiv);        ShowBigLine(0);
        delay_ms(ONE_SECOND);

```



```

sprintf(buff," ЗАЖИГАННИЕ ");      ShowBigLine(3);
VALVE_U_REL=1;                      //вмикаємо запалення OUT7
delay_ms(ONE_SECOND);
////////////////////////////////////
VALVE_U_REL=1; //Подаем сигнал на удержание заслонки и клапана
if(z==0)
{
    sprintf(buff," ЗАСЛОНКА-> "); ShowBigLine(3);
    VALVE_U_REL=0; //відключаємо імпульс закритої заслонки
}
else
{
    sprintf(buff," КЛАПАН -> "); ShowBigLine(3);
}
delay_ms(ONE_SECOND);
////////////////////////////////////
////////
// !!!!добавить проверку выходного напряжения
if(rotor<3){STARTER_REL=1;}; // запускаємо стартер ,якщо ротор
не працює v1.4)
// !!!!добавить проверку выходного напряжения
temp=t3;
while (temp!=0)
{
    sprintf(buff," СТАРТЕР:%1.1s ",temp); ShowBigLine(3);
    delay_ms(ONE_SECOND);
    temp--;
    if(rotor>7){temp=0;}; // если двигатель запустился - переходим к
заслонке
};

```

					ІАЛЦ.467100.009 ДЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

```

STARTER_REL=0; //відключаємо стартер OUT6 на (1-
6с)

////////////////////////////////////
temp=t4; // прогріваємо генератор із заслонкою (5-30с)
while (temp!=0)
{
    sprintf(buff,"ПРОГРЕВ:%2.uc ",temp); ShowBigLine(3);
    delay_ms(ONE_SECOND);
    temp--;
    //!!! if(T>TminZ){temp=0;}; //якщо температура висока -
зупиняємо прогрів
};
////////////////////////////////////
VALVE_U_REL=0; //відключаємо імпульс закритої заслонки АБО
КЛАПАНА
if(z==0)
{
    sprintf(buff," ЗАСЛОНКА < "); ShowBigLine(3);
    delay_ms(HALF_SECOND); //пауза щоб не подать
встречное напряжение
    VALVE_D_REL=1; //відкриваємо заслонку OUT4 Реверсивный
импульс!!!
}
else
{
    sprintf(buff," КЛАПАН <- "); ShowBigLine(3);
    delay_ms(ONE_SECOND);
}
delay_ms(ONE_SECOND);

```

VALVE_D_REL=0; //відключаємо реверсивний імпульс для заслонки
и КЛАПАНА (на всякий випадок)

////////////////////////////////////

////!! /// додати перевірку (можливо вже завантажено?)

temp=RAZGON;

while (temp!=0){

 sprintf(buff," РАЗГОН:%2.0c ",temp); ShowBigLine(3);

 delay_ms(ONE_SECOND);

 temp--;

};

////////////////////////////////////

if(KilkistZapuskiv>0){ KilkistZapuskiv--;};

PositionPointer=31; // Перехід на перевірку і прогрів

break;

////////////////////////////////////

case 31: // ПЕРЕВІРКА РОБОТИ ГЕНЕРАТОРА ,ПРОГРІВ

LCD_Clear();

if(rotor<3){//ротор зупинився (генератор заглов)

 ON_REL=0; //вимикаємо запалення

 if(KilkistZapuskiv>0){

 sprintf(buff," ПОВТОРНЫЙ "); ShowBigLine(3);

 sprintf(buff," ЗАПУСК "); ShowBigLine(5);

 delay_ms(NEXT_ZAPUSK_SECOND);

 if(KilkistZapuskiv==1){delay_ms(NEXT_ZAPUSK_SECOND);};//

якщо остання спроба - то акумулятор відпочиває удвічі довше (10с)

 PositionPointer=30;

 }

 else //кількість запусків вичерпана

 {

					ІАЛЦ.467100.009 ДЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

```

        PositionPointer=40;    //переходимо на аварію "генератор не
запустився"
        goto begin;
    };
}
else //Генератор запустився, чекаємо підключення нагрзуки
{
    if(T<TminN){
        temp=t5;    // Мінімальний час до підключення
нагрзуки
        while (temp!=0)
        { //якщо генератор запустився , то чекаємо (Мінімальний час до
підключення нагрзуки)
            sprintf(buff,"ПРОГРЕВ:%2.uc ",temp); ShowBigLine(3);
            delay_ms(ONE_SECOND);
            temp--;
        };
    };
    //ТУТ!!! Можна підключити генератор (температура нормальна,
або час витримано)
    //але лише якщо вхідний автомат включений, та генератор видає
напругу
    //////////////////////////////////////
    if(GEN_OUT_PIN==1){ //на виході генератора немає
напруги? (лінія інверсна: 0 - напруга Є)
        ON_REL=0;        //вимикаємо запалення
        PositionPointer=41;    //переходимо на аварію "генератор не
видає напруги"
        enter=2;
        goto begin;

```

					ІАЛЦ.467100.009 ДЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

```

};
PerehodNaGenerator(); //підключення нагртки на генератор
PositionPointer=32; // Робочий режим
//      TimeSec=podsvet; PORTD.7=1; //вмикаємо підсвічування РКІ та
переходимо до РОБОТИ
};
break;
////////////////////////////////////

case 32: // РОБОТА (РУЧНА або АВТОМАТИЧНА)
    sprintf(buff," РАБОТА  ");      ShowBigLine(0);
    MainInfo();

    if(AvtRu==0)//якщо робота АВТОМАТИЧНА то потрібен контроль
появи струму!!
    {
//!!!ПОМЕНЯТЬ
        if((U1>150)&&(U1<250))
        { //напруга в мережі з'явилась?
            //////////////////////////////////
            temp=t2;      // завантажуюємо затримку переходу на основну
мережу

            while (temp!=0) // ждем, вдруг сеть пропадет снова?
            {
                sprintf(buff," СЕТЬ НОРМА "); ShowBigLine(3);
                sprintf(buff,"  %3.uc  ",temp);ShowBigLine(5);
                delay_ms(ONE_SECOND);
                temp--;
            }
//!!!ПОМЕНЯТЬ

```

```

/*2.2*/      if ((U1>140)&&(U1<255)){PositionPointer=33;enter=2;goto
begin;} //напруга все ще є?,тоді перехід.
    }
}
else // ручная работа- тогда останов кнопкой!
{
    if(enter>GO_TO_OPTIONS){PositionPointer=33;enter=2;};
}

if(GEN_OUT_PIN==1){ //на виході генератора зникла напруга? (лінія
інверсна: 0 - напруга Є)
    PerehodNaMereju();
    ON_REL=0;          //вимикаємо запалення
    PositionPointer=41; //переходимо на аварію "генератор не видає
напруги"

    enter=2;
    LCD_Clear();

};
break;

////////////////////////////////////

case 33: // ОСТАНОВ С ОХЛАЖДЕНИЕМ
    sprintf(buff," ОСТАНОВ ");      ShowBigLine(0);
    PerehodNaMereju();
    delay_ms(ONE_SECOND);
    //////////////////////////////////
    temp=t5*2; //завантажуємо час охолодження
    while (temp!=0) // ждем, вдруг сеть пропадет снова?
    {
        sprintf(buff," ОХЛАЖДЕНИЕ "); ShowBigLine(3);
        sprintf(buff,"  %3.uc  ",temp);ShowBigLine(5);
    }

```

```

delay_ms(ONE_SECOND);
temp--;
if(AvtRu==0)
{ //якщо робота АВТОМАТИЧНА то потрібен контроль
ЗНИКНЕННЯ струму!!
    if ((U1<140)||(U1>265))
    { //напруга в мережі знову зникла
        sprintf(buff," НЕТ СЕТИ! "); ShowBigLine(3);
        sprintf(buff,"  %3.uc  ",temp);ShowBigLine(5);
        delay_ms(ONE_SECOND);
        PerehodNaGenerator(); //підключення нагрздки на генератор
        PositionPointer=32; // перехід у "Робочий режим"
        enter=2; goto begin;
    }
}
}
}
////////////////////////////////////
temp=5; // завантажуюмо час перекриття паливного клапану (при
зупинці)
OFF_REL=1; // перекриваємо паливний клапан
while (temp!=0){
//    sprintf(buff," ТОПЛИВНЫЙ "); ShowBigLine(3);
    sprintf(buff," Т.КЛАП%2.uc",temp);ShowBigLine(5);
    delay_ms(ONE_SECOND);
    // PORTD.6=0;//вимикаємо запалення после первой секунды
роботы клапана
    temp--;
};
OFF_REL=0; // відкриваємо паливний клапан
////////////////////////////////////

```

					ІАЛЦ.467100.009 ДЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

```

        if(AvtRu==0) //для автоматичного режиму - повернення у режим
АВТОМАТ  інакше в ИНФО
    {
        PositionPointer=1;
    }
    else
    {
        PositionPointer=3;
    }
    ON_REL=0;//вимикаємо запалення
    goto begin;
break;

/////////////////////////////////////////////////////////////////

case 40: // АВАРИЯ ГЕНЕРАТОР НЕ ЗАПУСТИЛСЯ
    sprintf(buff, "(!) ОШИБКА ");      ShowBigLine(0);
    sprintf(buff, " НЕ ЗАПУСК ");      ShowBigLine(3);
//    sprintf(buff, "    ДЛЯ СБРОСА    "); ShowLine(INFOLINE1);
//    sprintf(buff, "    УДЕРЖИВАЙ 'ВВОД' "); ShowLine(INFOLINE2);
    sprintf(buff, "    ДЛЯ СБРОСА >'ВВОД' "); ShowLine(INFOLINE1);
    delay_ms(ONE_SECOND);
    LCD_Blinc(QUARTER_SECOND,3);
    if(enter>GO_TO_OPTIONS){PositionPointer=LAST_MODE_MENU;ente
r=2;LCD_Clear();}
    break;

/////////////////////////////////////////////////////////////////

case 41: // АВАРИЯ НЕТ ТОКА ОТ ГЕНЕРАТОРА
    sprintf(buff, "(!) ОШИБКА ");      ShowBigLine(0);
    sprintf(buff, " НЕТ ТОКА ");      ShowBigLine(3);
    sprintf(buff, "    ДЛЯ СБРОСА >'ВВОД' "); ShowLine(INFOLINE1);

```



```

delay_ms(ONE_SECOND);

LCD_Blinc(QUARTER_SECOND,3);

if(enter>GO_TO_OPTIONS){PositionPointer=LAST_MODE_MENU;enter=2;LCD_Clear();}

break;

////////////////////////////////////

case 42: // АВАРИЯ НЕТ ТОКА ОТ ГЕНЕРАТОРА

    LINE_REL=1; // відключаємо МЕРЕЖУ   !!!
    GEN_REL=0; // відключаємо ГЕНЕРАТОР
    ON_REL =0; //вимикаємо: запалення OUT7
    VALVE_U_REL=0; //заслонку OUT5 (Port4)
    VALVE_D_REL=0;
    STARTER_REL=0; //вимикаємо стартер OUT6


    sprintf(buff,"(!) ЗАЩИТА ");    ShowBigLine(0);
    if(U1<255){
        sprintf(buff," Uin =%4.uB",U1);    ShowBigLine(3);
        sprintf(buff," ДЛЯ СБРОСА >'ВВОД' "); ShowLine(INFOLINE1);
    }
    else
    {
        sprintf(buff,"!Uin > 260В!");    ShowBigLine(3);
        sprintf(buff," ВЫСОКОЕ НАПРЯЖЕНИЕ !"); ShowLine(INFOLINE1);
//    sprintf(buff,"    СЕТИ !    "); ShowLine(INFOLINE2);
    }

    delay_ms(ONE_SECOND);

    LCD_Blinc(QUARTER_SECOND,3);

```

```

        if((U1<255)&(enter>GO_TO_OPTIONS)){PositionPointer=LAST_MODE
_MENU;PORTD.0=0;enter=2;LCD_Clear();} // PORTD.0=0 - підключаємо
МЕРЕЖУ

        break;

////////////////////////////////////

    };// switch (PositionPointer)

begin:

    } //while

} //main

```

					ІАЛЦ.467100.009 ДЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		30

Додаток 4

Контролер керування автоматичним запуском побутових електростанцій

Код програми дистанційного керування генератором.

Аркушів 8

```

// БИБЛИОТЕКИ

#include    <iarduino_GSM.h>

// Подключаем библиотеку iarduino_GSM для работы с GSM/GPRS Shield.

#include    <SoftwareSerial.h>

// Подключаем библиотеку SoftwareSerial для программной реализации шины
UART.

iarduino_GSM  gsm;

// Создаём объект gsm для работы с функциями и методами библиотеки
iarduino_GSM.

SoftwareSerial  softSerial(7, 8);

// Создаём объект softSerial для работы по программной шине UATR,
указывая выводы RX и TX платы Arduino (выводы 7 и 8)

// ПЕРЕМЕННЫЕ

uint8_t      Sum      = 0;

// Переменная для хранения процентного значения ШИМ-сигнала
// Для отправки СМС о удачном запуске устройства, укажите в строке ниже
номер телефона без пробелов.

char          SMSnum[13] = "7_ВАШ_НОМЕР_ТЕЛЕФОНА";

// Объявляем строку для хранения номера отправителя SMS сообщений.

char          SMStxt[161];

// Объявляем строку для хранения текста принятых SMS сообщений.

String        strSMStxt;

// переменная типа String для более удобной работы

uint8_t       flgFunc;

// флаг наличия действия

uint8_t       flgDev;

// флаг наличия устройства

```

					ІАЛЦ.467100.010 Д4			
Зм.	Арк.	№ докум.	Підпис	Дата				
Розробив		Баранник Д.Є.			Контролер керування автоматичним запуском побутових електростанцій Код програми дистанційного керування генератором.	Літ.	Аркуш	Аркушів
Перевір.		Таран В. І.					1	8
						КПІ ім. Ігоря Сікорського Каф. ФІОТ, гр. ІО-361		
Н. контр.		Сімоненко В.П.						
Затверд.								

```

uint8_t    flgPWM;
// флаг наличия ШИМ
uint8_t    lvlPWM;
// Переменная количества символов в строке
const char*  Func[2][5] = {{"ВКЛЮЧИТЬ", "ВЫКЛЮЧИТЬ"}, {"СВЕТ
КУХНЯ", "СВЕТ КОМНАТА", "ДВЕРЬ", "ВЕНТИЛЯТОР",
"ПОДСВЕТКА"}};
// Двумерный массив с функциями и устройствами
const char  Pins[]  = { 2,      3,      4,      5,      6  };
// Массив с номерами выводов устройств
void setup()
{
// Иницилируем GSM/GPRS Shield и проверяем его готовность к работе:
gsm.begin(softSerial);
// Иницилируем работу GSM/GPRS Shield, указывая объект шины UART.
while (gsm.status() != GSM_OK) { delay(1000); }
// Ждём завершения регистрации модема в сети оператора связи.
// Установка кодировки для символов Кириллицы:
gsm.TXTsendCodingDetect("п");
// Выполняем автоопределение кодировки скетча для отправки текста на
Русском языке.
// Отправка сообщения об удачном запуске:
gsm.SMSsend( F("Инициализация прошла успешно."), SMSnum);
// Данная строка будет отвечать отправителю указанным SMS сообщением.
// Настраиваем выводы
pinMode(Pins[0], OUTPUT);

```

```

// настройка вывода на работу в режиме выхода
pinMode(Pins[1], OUTPUT);
// настройка вывода на работу в режиме выхода
pinMode(Pins[2], OUTPUT);
// настройка вывода на работу в режиме выхода
pinMode(Pins[3], OUTPUT);
// настройка вывода на работу в режиме выхода
pinMode(Pins[4], OUTPUT);
// настройка вывода на работу в режиме выхода
digitalWrite(Pins[0], LOW);
// устанавливаем на выходе значение сигнала равным LOW
digitalWrite(Pins[1], LOW);
// устанавливаем на выходе значение сигнала равным LOW
digitalWrite(Pins[2], LOW);
// устанавливаем на выходе значение сигнала равным LOW
digitalWrite(Pins[3], LOW);
// устанавливаем на выходе значение сигнала равным LOW
digitalWrite(Pins[4], LOW);
// устанавливаем на выходе значение сигнала равным LOW
flgFunc = 20;    // Сбрасываем флаг
flgDev  = 20;    // Сбрасываем флаг
flgPWM  = 0;    // Сбрасываем флаг
}
void loop ()
{
if (millis() % 1000 < 100) {
// Выполняем код в теле оператора if первые 100 мс каждой секунды.

```

```

delay(100);

// Устанавливаем задержку в 100 мс, чтоб не выполнить код более 1 раза за
секунду.

    if (gsm.SMSavailable()) {
// Функция SMSavailable() возвращает количество входящих непрочитанных
SMS сообщений.

        gsm.SMSread(SMStxt, SMSnum);
// Читаем SMS сообщение в ранее объявленные переменные (текст SMS
сообщения, адрес отправителя, дата отправки, идентификатор SMS,
количество SMS, номер SMS).

        strSMStxt = SMStxt;
// Присваиваем массиву atrSMStxt значения массива SMStxt для дальнейшей
проверки на совпадения

        for (int m = 0; m < (sizeof(Func[0]) / 2); m++) {
// проверяем в цикле, совпадает ли команда из массива с тем, что пришло в
СМС

            if (strSMStxt.indexOf(Func[0][m]) > -1 && *Func[0][m] != 0) {
// если совпадения есть, то

                flgFunc = m;
// присваиваем флагу порядковый номер совпавшего слова из массива Func

                for (int i = 0; i < (sizeof(Func[1]) / 2); i++) {
// проверяем в цикле, совпадает ли имя устройства из массива с тем, что
пришло в СМС

                    if (strSMStxt.indexOf(Func[1][i]) > -1 && *Func[1][i] != 0) {
// если совпадения есть и это не пустая строка, то

                        flgDev = i;
// присваиваем флагу порядковый номер совпавшего слова из массива Device

```

```

        if (strchr(SMStxt, '%')) {
// проверяем, есть ли в тексте СМС символ %
            flgPWM = 1;
// и если есть, то устанавливаем флаг
        }
    }
}
}
}

if (flgPWM == 1) {
// если установлен флаг flgPWM, то
    lvlPWM = uint8_t(strchr(SMStxt, '%' ) - SMStxt);
// определяем количество символов в строке до совпадения с символом %
    if (SMStxt[lvlPWM - 1] >= '0' && SMStxt[lvlPWM - 1] <= '9') {
// Проверяем, является ли символ, расположенный в строке на 1 позицию
// левее символа %, цифрой от 0 до 9, и если да, то
        Sum = uint8_t(SMStxt[lvlPWM - 1] - '0');
// прибавляем эту цифру к переменной (единицы)
        if (SMStxt[lvlPWM - 2] >= '0' && SMStxt[lvlPWM - 2] <= '9') {
// Проверяем, является ли символ, расположенный в строке на 2 позиции
// левее символа %, цифрой от 0 до 9, и если да, то
            Sum += uint8_t((SMStxt[lvlPWM - 2] - '0') * 10);
// прибавляем эту цифру к переменной, умножив на 10 (десятки)
            if (SMStxt[lvlPWM - 3] > '0' && SMStxt[lvlPWM - 3] <= '1') {
// Проверяем, является ли символ, расположенный в строке на 3 позиции
// левее символа %, цифрой от 0 до 1, и если да, то
                Sum += uint8_t((SMStxt[lvlPWM - 3] - '0') * 100) ;

```



```

// прибавляем эту цифру к переменной, умножив на 100 (сотни)
    } else      // иначе проверяем, что
        if (SMStxt[lvlPWM - 3] >= '2' && SMStxt[lvlPWM - 3] <= '9') {
// число "сотен" не больше 1, и если больше то
        flgPWM = 0; // сбрасываем флаг ШИМ
        }
    }
}
}

if (flgFunc != 20) { // если флаг flgFunc установлен, то
    switch (flgFunc) {
// проверяем значение флага flgFunc с определёнными в операторе case
        case 0:      // Если 0 - значит указано действие ВКЛЮЧИТЬ
            if (flgDev != 20) { // Если установлен флаг наличия устройства, то
                if (flgPWM == 1) { // проверяем флаг ШИМ. Если он установлен,
то
                    uint8_t mapPWM = map(Sum, 0, 100, 0, 255);
// переопределяем множество для ШИМ и
                    analogWrite(Pins[flgDev], mapPWM ); // подаём ШИМ-сигнал
на вывод и
                    gsm.SMSsend(F("Процент задан!"), SMSnum);
// отправляем сообщение об успешном выполнении.
                    Sum = 0;
// сбрасываем количество процентов для задания ШИМ-сигнала после
выполнения
                    flgPWM = 0; // сбрасываем флаг ШИМ после выполнения
                }
            }
        }
    }
}

```

					ІАЛЦ.467100.010 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

```

else {    // Если же флага ШИМ нет, то
    digitalWrite(Pins[flgDev], HIGH);
// подаём высокий сигнал на вывод для включения и
    gsm.SMSsend(F("Устройство включено!"), SMSnum);
// отправляем сообщение об успешном выполнении.
}
flgDev = 20;    // сбрасываем флаг устройства после выполнения
} else {    // если в СМС не указано устройство, то
    gsm.SMSsend(F("Что включить?"), SMSnum);
// отправляем об этом сообщение
}
flgFunc = 20;    // сбрасываем флаг действия после выполнения
break;

case 1:    // Если 1 - значит указано действие ВЫКЛЮЧИТЬ
    if (flgDev != 20) {    // Если установлен флаг наличия
устройства, то
        digitalWrite(Pins[flgDev], LOW);
// подаём низкий сигнал на вывод для выключения
        gsm.SMSsend(F("Устройство выключено!"), SMSnum);
// и отправляем сообщение об успешном выполнении.
        flgDev = 20;    // сбрасываем флаг устройства после выполнения
    } else {    // если в СМС не указано устройство, то
        gsm.SMSsend(F("Что выключить?"), SMSnum);
// отправляем об этом сообщение
    }
    flgFunc = 20;    // сбрасываем флаг действия после выполнения
    break;

```

```

    }
    } else {
        gsm.SMSsend(F("Ошибка в тексте!"), SMSnum);
    // Если текст пришёл не верный, то отправляем об этом сообщение
    }
    }
    }
    }
    }

```

					ІАЛЦ.467100.010 Д4	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		